Fuzzy Expert Systems

Jonathan M. Garibaldi

Automated **S**cheduling, Optimis**A**tion and **P**lanning (ASAP) Research Group
School of Computer Science and IT, University of Nottingham,
Jubilee Campus, Wollaton Road, Nottingham, UK, NG8 1BB
jmg@cs.nott.ac.uk

In this Chapter, the steps necessary to develop a fuzzy expert system (FES) from the initial model design through to final system evaluation will be presented. The current state-of-the-art of fuzzy modelling can be summed up informally as 'anything goes'. What this actually means is that the developer of the fuzzy model is faced with many steps in the process each with many options from which selections must be made. In general, there is no specific or prescriptive method that can be used to make these choices, there are simply heuristics ('rules-of-thumb') which may be employed to help guide the process. Each of the steps will be described in detail, a summary of the main options available will be provided and the available heuristics to guide selection will be reviewed.

The steps will be illustrated by describing two cases studies: one will be a mock example of a fuzzy expert system for financial forecasting and the other will be a real example of a fuzzy expert system for a medical application. The expert system framework considered here is restricted to rule-based systems. While there are other frameworks that have been proposed for processing information utilising fuzzy methodologies, these are generally less popular in the context of fuzzy expert systems.

As a note on terminology, the term *model* is used to refer to the abstract conception of the process being studied and hence *fuzzy model* is the notional representation of the process in terms of fuzzy variables, rules and methods that together define the input-output mapping relationship. In contrast, the term *system* (as in fuzzy expert system) is used to refer to the embodiment, realisation or implementation of the theoretical model in some software language or package. A single model may be realised in different forms, for example, via differing software languages or differing hardware platforms. Thus it should be realised that there is a subtle, but important, distinction between the evaluation of a fuzzy model of expertise and the evaluation of (one or more of) its corresponding fuzzy expert systems. A model may be evaluated as accurately capturing or representing the domain problem under consideration, whereas its realisation as software might contain bug(s) that cause undesired artefacts in the output. This topic will be further explored in Sect. 7.

It will generally be assumed that the reader is familiar with fuzzy theory, methods and terminology — this Chapter is *not* intended to be an introductory tutorial, rather it is a guide to currently accepted best practice for building a fuzzy expert system. For a simple introductory tutorial the reader is referred to Cox [8]; for a comprehensive coverage of fuzzy methods see, for example, Klir and Yuan [22], Ruspini et al [34] or Kasabov [21].

The central question for this Chapter is 'what are smart adaptive fuzzy expert systems?' In current state-of-the-art it is *not* possible to automatically adapt a system created in one application area to address a novel application area. Automatic tuning or optimisation techniques may be applied to (in some sense) *adapt* a given fuzzy expert system to particular data (see Sect. 8). Real 'smart adaptive' systems are presently more likely to be found in neuro-fuzzy or hybrid systems covered in subsequent Chapters. In each new application area, a fuzzy expert system must effectively be 'hand-crafted' to achieve the desired performance. Thus the creation of good fuzzy expert systems is an art that requires skill and experience. Hints and tips to assist those new to this area in making appropriate choices at each stage will be provided.

1 Introduction

The generic architecture of a fuzzy expert system showing the flow of data through the system is shown in Fig. 1 (adapted from Mendel [26]). The general process of constructing such a fuzzy expert system from initial model design to system evaluation is shown in Fig. 2. This illustrates the typical process flow as distinct stages for clarity but in reality the process is not usually composed of such separate discrete steps and many of the stages, although present, are blurred into each other.

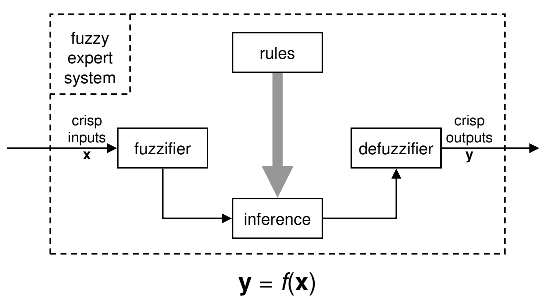

Fig. 1. Generic architecture of a fuzzy expert system

Once the problem has been clearly specified (see Chap. 2), the process of constructing the fuzzy expert system can begin. Invariably some degree of data

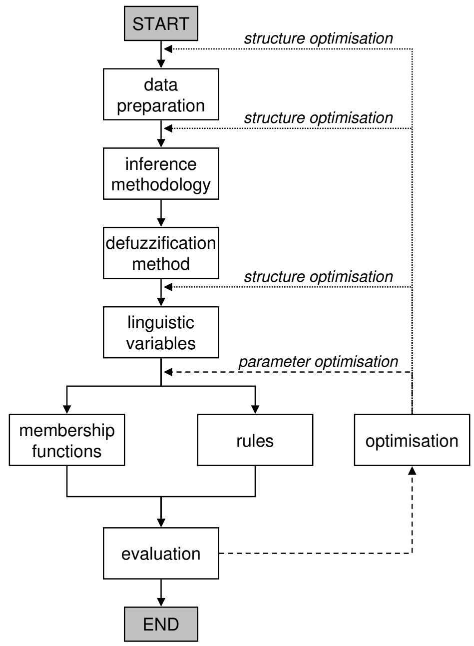

Fig. 2. Typical process flow in constructing a fuzzy expert system

preparation and preprocessing is required, and this stage has been discussed in detail in Chap. 3. The first major choice the designer has to face is whether to use the Mamdani inference method [24] or the Takagi-Sugeno-Kang (TSK) method [37, 39]. The essential difference in these two methodologies is that the result of Mamdani inference is one or more fuzzy sets which must (almost always) then be defuzzified into one or more real numbers, whereas the result of TSK inference is one or more real functions which may be evaluated directly. Thus the choice of inference methodology is linked to the choice of defuzzification method. Once the inference methodology and defuzzification method have been chosen, the process of enumerating the linguistic variables necessary can commence. This should be relatively straightforward if the problem has been well specified and is reasonably well understood. If this is not the case, then the decision to construct a fuzzy expert system may not be appropriate. The next stage of deciding the necessary terms with their defining membership functions and determining the rules to be used is far from trivial however. Indeed, this stage is usually the most difficult and time consuming of the whole process.

After a set of fuzzy membership functions and rules has been established the system may be evaluated, usually by comparison of the obtained output against some *desired* or *known* output using some form of error or distance function. However, it is very rare that the first system constructed will perform at an acceptable level. Usually some form of optimisation or performance tuning of the system will need to be undertaken. Again, there are a multitude of options that a designer may consider for model optimisation. A primary distinction illustrated in Fig. 2 is the use of either *parameter optimisation* in which (usually) only aspects of the model such as the shape and location of membership functions and the number and form of rules are altered, or *structure optimisation* in which all aspects of the system including items such as the inference methodology, defuzzification method, or number of linguistic variables may be altered. In general, though, there is no clear distinction. Some authors consider rule modification to be structure optimisation, while others parameterise the rules.

1.1 Case Studies

As stated earlier, two case studies will be used through the rest of this Chapter to provide a grounding for the discussions. The two case studies are now briefly introduced.

Financial Forecasting Fuzzy Expert System

The problem is to design a fuzzy expert system to predict (advise) when to buy or sell shares in an American company based on three sources of information:

1. the past share price of the company itself (*share_pr*),
2. the Euro / Dollar exchange rate (*xchg_rate*), and
3. the FTSE (Financial Times Stock Exchange) share index (the UK stock index) ($FTSE$).

Clearly, this is an artificial scenario as the Dow Jones index would almost certainly be used in any real predictor of an American company, but the case study is designed to be illustrative rather than realistic.

Umbilical Acid-Base Fuzzy Expert System

Childbirth is a stressful experience for both mother and infant. Even during normal labour every infant is being regularly deprived of oxygen as maternal contractions, which increase in frequency and duration throughout labour until delivery, restrict blood supply to the placenta. This oxygen deprivation can lead to fetal 'distress', permanent brain damage and, in the extreme, fetal death. An assessment of neonatal status may be obtained from analysis of blood in the umbilical cord of an infant immediately after delivery. The umbilical cord vein carries blood from the placenta to the fetus and the two smaller cord arteries return blood from the fetus. The blood from the placenta has been freshly oxygenated, and has a relatively high partial pressure of oxygen ($p\mathrm{O}_2$) and low partial pressure of carbon dioxide ($p\mathrm{CO}_2$). Oxygen in the blood fuels *aerobic* cell metabolism, with carbon dioxide produced as 'waste'. Thus the blood returning from the fetus has relatively low oxygen and high carbon dioxide content. Some carbon dioxide dissociates to form carbonic acid in the blood, which increases the acidity (lowers the pH). If oxygen supplies are too low, *anaerobic* (without oxygen) metabolism can supplement aerobic metabolism to maintain essential cell function, but this produces lactic acid as 'waste'. This further acidifies the blood, and can indicate serious problems for the fetus.

A sample of blood is taken from each of the blood vessels in the clamped umbilical cord and a blood gas analysis machine measures the pH, $p\mathrm{O}_2$ and $p\mathrm{CO}_2$. A parameter termed *base deficit of extracellular fluid* ($\mathrm{BD}_{\mathrm{ecf}}$) can be derived from the pH and $p\mathrm{CO}_2$ parameters [35]. This can distinguish the cause of a low pH between the distinct physiological conditions of *respiratory acidosis*, due to a short-term accumulation of carbon dioxide, and a *metabolic acidosis*, due to lactic acid from a longer-term oxygen deficiency. An interpretation of the status of health of an infant can be made based on the pH and $\mathrm{BD}_{\mathrm{ecf}}$ parameters ('the acid-base status') of both arterial and venous blood. However, this is difficult to do and requires considerable expertise.

A fuzzy expert system was developed for the analysis of umbilical cord acid-base status, encapsulating the knowledge of leading obstetricians, neonatologists and physiologists gained over years of acid-base interpretation. The expert system combines knowledge of the errors likely to occur in acid-base measurement, physiological knowledge of plausible results, statistical analysis of a large database of results and clinical experience of acid-base interpretation. It automatically checks for errors in input parameters, identifies the vessel origin (artery or vein) of the results and provides an interpretation in an objective, consistent and intelligent manner. For a full description of the development and evaluation of this fuzzy expert system, see [11, 12, 14, 15].

2 Data Preparation and Preprocessing

Data preparation and preprocessing has already been discussed in detail in Chap. 3. Issues such as feature selection, normalisation, outlier removal, etc., are as important to fuzzy expert systems as to any other systems. The reason data preparation and preprocessing is mentioned again here is that it *must* be considered as an integral part of the modelling process, rather than being a fixed procedure carried out prior to modelling. This implies that data preparation methods may be included as components of model structure that are to be optimised in some way.

As an example, consider the financial forecasting FES. It would be unusual and of little obvious value to use the *absolute* values of any of the three primary variables (*share_pr*, *xchg_rate* and *FTSE*) as input to the FES. A more likely choice would be to use difference information such as daily, weekly or monthly differences. So, for example, an FES could reasonably be imagined which took nine input variables (the daily, weekly and monthly differences of *share_pr*, *xchg_rate* and *FTSE*) and produced one output variable *advice*. Alternatively, it might be that all useful information was contained in daily differences so that only three input variables were used. Viewed this way, an optimisation technique might be employed to carry out combinatorial optimisation from a fixed choice of input variable combinations. Alternatively, the period (no. of days) over which the difference is calculated could be included as a tunable parameter within data preparation which could be carried out under control of a parameter optimisation technique.

From the above it can be seen that data preparation is also closely linked to the choice of linguistic variables for the FES. The most important aspect to be considered is that the universe of discourse (range) of each linguistic variable should be fixed. This is the reason that the absolute values of *share_pr*, *xchg_rate* and *FTSE* are unlikely to be used in a financial forecasting FES. If the universe of discourse of each variable is not fixed in advance, then the FES would have to self-adapt to new ranges. This is a current area of active research, but solutions to this problem have not yet been found.

2.1 Hints and Tips

- Data preparation and preprocessing is an essential and fundamental component of fuzzy expert system design. It *must* be considered as another component of the process that can be adjusted, altered or optimised as part of creating a successful fuzzy expert system.
- To paraphrase Einstein, data preparation and preprocessing should be as 'simple as possible, but no simpler'. Always *consider* utilising raw data as input to the FES where possible (although this may not be possible in situations such as in the presence of high noise levels).

3 Inference Methodology

3.1 Mamdani Inference

In Mamdani inference [24], rules are of the following form:

$$\begin{gathered} R_i \quad \text{if } x_1 \text{ is } A_{i1} \text{ and } \ \ldots \ \text{ and } x_r \text{ is } A_{ir} \text{ then } y \text{ is } C_i \\ \text{for } i = 1, 2, \ \ldots, \ L \end{gathered} \tag{1}$$

where L is the number of rules, x_j $(j = 1, 2, \ldots, r)$ are the input variables, y is the output variable, and A_{ij} and C_i are fuzzy sets that are characterised by membership functions $A_{ij}(x_j)$ and $C_i(y)$, respectively. The important thing to note is that the consequence of each rule is characterised by a fuzzy set C_i. An example of Mamdani inference for an illustrative three-rule system is shown in Fig. 3. The final output of a Mamdani system is one or more arbitrarily complex fuzzy sets which (usually) need to be defuzzified.

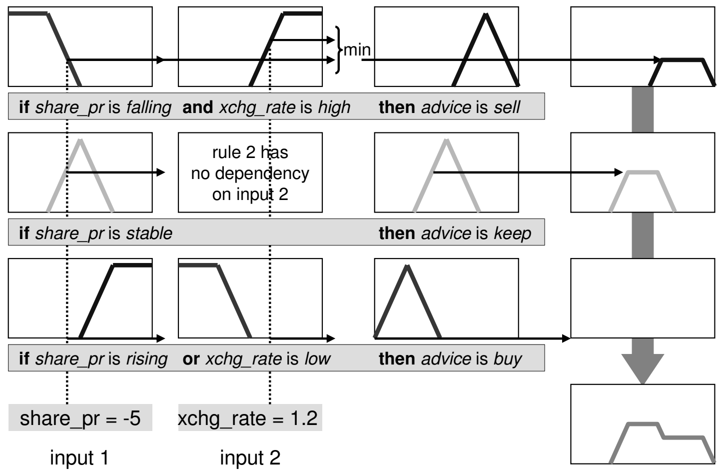

Fig. 3. A three-rule Mamdani inference process

In the context of umbilical acid-base analysis described earlier, the final expert system comprised a set of Mamdani inference rules operating on four input variables — the acidity and base-deficit of the arterial and venous blood (pH_A, BD_A, pH_V and BD_V) — to produce three output variables, severity of *acidemia* (ranging from severely acidemic to alkolotic), *component* of acidemia (ranging from pure metabolic, through mixed, to pure respiratory) and *duration* of acidemia (ranging from chronic to acute).

3.2 Takagi-Sugeno-Kang Inference

In Takagi-Sugeno-Kang (TSK) inference [37, 39], rules are of the following form:

$$
\begin{aligned}
R_i \qquad & \text{if } x_1 \text{ is } A_{i1} \text{ and } \ \ldots \ \text{ and } x_r \text{ is } A_{ir} \\
& \text{then } y_i = b_{i0} + \ b_{i1}x_1 + \ \ldots \ + \ b_{ir}x_r \\
& \qquad\qquad \text{for } i = 1, 2, \ \ldots, \ L
\end{aligned}
\tag{2}
$$

where b_{ij} $(j = 1, 2, \ldots, r)$ are real-valued parameters. The important thing to note here is that the consequence of each rule is now characterised by a linear function of the original input variables. The final output the inference process is calculated by:

$$
y = \frac{\sum_{i=1}^{L} \alpha_i (b_{i0} + \ b_{i1}x_1 + \ \ldots \ + \ b_{ir}x_r)}{\sum_{i=1}^{L} \alpha_i}
\tag{3}
$$

where α_i is the firing strength of rule R_i. The *order* of a TSK system is the degree of (highest exponent of x in) the consequent function. The most commonly used TSK model is the first-order model. An example of zeroth-order TSK inference for an illustrative three-rule system is shown in Fig. 4.

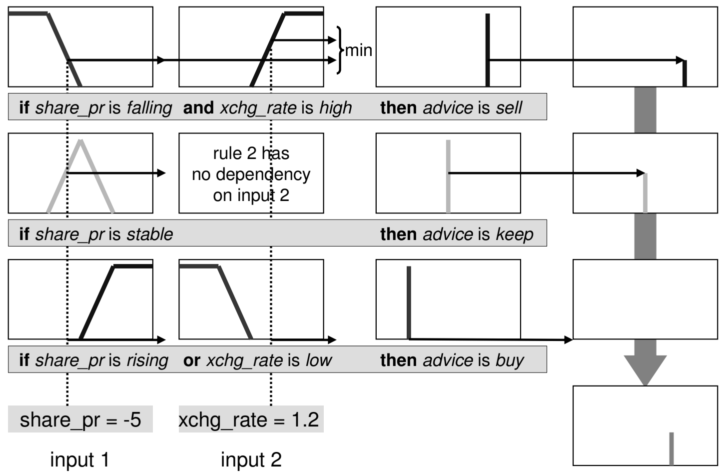

Fig. 4. A three-rule zeroth-order TSK inference process

3.3 Choice of Inference Method

Which method should be chosen? There is no simple answer to this question; rather, the answer depends on the nature of the expert system being developed. Two more specific questions can be framed: (i) Is any expression of uncertainty or non-numeric information required in the answer? (ii) Is processing speed (e.g. real-time operation) or memory usage crucial? If the answer to the first is 'yes', then Mamdani inference is clearly favoured. If the answer to the second is 'yes', then TSK inference is clearly favoured. Otherwise, the choice is a matter of preference, although in many cases, TSK based methods may result in fewer rules. As an example, consider the two case-studies presented. In the financial forecasting example, there would appear to be little / no need for uncertainty representation in the output. If the output is above a threshold, the advice will be to buy shares; if the output is below a threshold, the advice will be to sell shares held. Although there is no specific requirement for real-time operation, TSK inference might be utilised as a Mamdani system would just introduce an 'uncecessary' step of defuzzification. In the umbilical acid-base expert system, there was a specific requirement both for a representation of uncertainty in the output and for the potential of obtaining a linguistic rather than numeric output (see Sect. 6 for information on obtaining linguistic output). There was no real-time constraint and hence Mamdani inference was clearly indicated.

3.4 Fuzzy Operators

A great deal has been written about the theoretical properties and practical choices of operators necessary to carry out fuzzy operations of set intersection

(AND) and union (OR). It is well established that fuzzy intersections are represented by a class of functions that are called *triangular norms* or *T-norms* and that fuzzy unions are represented by *triangular conorms* or *T-conorms*. A T-norm $\otimes$ is a binary function:

$$\otimes : \mathbb{I}^2 \rightarrow \mathbb{I}$$

where $\mathbb{I}$ represents the set of real numbers in the unit interval $[0, 1]$, that satisfies the following conditions: for any $a, b \in \mathbb{I}$,

$$\begin{array}{lll} (i) & 1 \otimes a = a & (\textit{identity}) \\ (ii) & (a \otimes b) \otimes c = a \otimes (b \otimes c) & (\textit{associativity}) \\ (iii) & a \otimes b = b \otimes a & (\textit{commutivity}) \\ (iv) & a \otimes b \leq a \otimes c, \text{ if } b \leq c & (\textit{monotonicity}) \end{array}$$

A number of operators and parameterised operator families have been proposed which satisfy these conditions, but by far the most common in practical use are either of:

$$\begin{array}{ll} a \otimes b = \min(a, b) & (\textit{standard}) \\ a \otimes b = ab & (\textit{algebraic product}) \end{array}$$

A T-conorm $\oplus$ is a similar binary function on $\mathbb{I}^2$ that satisfies the same conditions as a T-norm with the exception of:

$$(i) \quad 0 \oplus a = a \qquad (\textit{identity})$$

The corresponding T-conorms to the two T-norms specified above are:

$$\begin{array}{ll} a \oplus b = \max(a, b) & (\textit{standard}) \\ a \oplus b = a + b - ab & (\textit{algebraic sum}) \end{array}$$

In most fuzzy expert systems utilising Mamdani inference, the T-norm operator is used for the intersection operator to combine clauses in the rule antecedent joined with an *and*, and for the implication operator to combine the firing strength of the overall antecedent with the consequent fuzzy set. The T-conorm operator is used for the union operator to combine clauses in the rule antecedent joined with an *or*, and for the combination operator to aggregate the consequents of each rule into a single consequent set. The *min* T-norm and the *max* T-conorm are known as the *standard* operators as they were the original operators proposed by Zadeh [43]. They are the only T-norm and T-conorm that also obey the condition of *idempotence*: that is $a \otimes a = a$ and $a \oplus a = a$, respectively.

Several observations concerning current state-of-the-art of fuzzy operators can be made. Firstly, operators are generally used as a consistent family: i.e. if the standard *min* operator is used for intersection and implication, then the *max* operator should be used for union and aggregation, etc. Secondly, the performance differences that are obtained in real expert systems are often very small and of secondary importance compared to, for example, the membership functions of the terms or the rule set. Within the umbilical acid-base expert system a comparison was made of the performance difference between the standard operator family and the algebraic family. The algebraic family was found to increase performance (in comparison with human experts) by around 1% overall. Thirdly, in the context of the same expert system it was also observed that human experts involved in the model design had a definite subjective preference for the algebraic operator family. This appeared to be due to the fact that use of the standard *min* and *max* operators caused plateaus to appear in the consequent sets, as the consequent set is truncated at the height of the rule firing strength. Having become acquainted with the philosophies and notions of 'fuzzy logic', the human experts found such plateaus 'unnatural'. Of course, this subjective preference is possibly less important than the performance gain mentioned above.

3.5 Hints and Tips

- Use Mamdani inference if some expression of uncertainty in the output is required. An associated defuzzification technique is usually employed.
- Use TSK inference when a numeric output with no associated expression of uncertainty is required, or processing speed is essential.
- Fuzzy operators should be chosen as a consistent family: i.e. if *max* is used for the intersection operator, *min* should be used for union; similarly the *algebraic product* and *algebraic sum* should be used together.
- The choice of fuzzy operator family rarely makes a sizeable difference to the overall performance of a fuzzy expert system.

4 Linguistic Variables and Membership Functions

Obtaining an appropriate set of linguistic variables, their associated terms, and a set of rules lies at the heart of the creation of any fuzzy model of expertise. Rule determination will be discussed further in the next Section.

The choice of linguistic variables is primarily a combination of knowledge elicitation and data preparation. For example, in the umbilical acid-base FES there are six primary variables (arterial and venous pH, $p\mathrm{CO}_2$, and $p\mathrm{O}_2$) available, yet four linguistic variables (arterial and venous pH and $\mathrm{BD}_{\mathrm{ecf}}$) are used for interpretation, two of which (BD_{A} and BD_{V}) are derived. The reason is simply that knowledge elicitation sessions with domain experts indicated that these four variables are the only ones necessary. In the case of the financial forecasting FES, the question of which linguistic variables should be used is more difficult. In such a situation, the processes of data preparation, linguistic variable and membership function specification, rule determination and model optimisation merge into a single interdependent process.

The question of the number and names of the individual terms in each linguistic variable is also difficult. In this case, there are some practically accepted (but not theoretically justified) heuristics. The terms of a linguistic variable should be:

- justifiable in number
  - the number of terms should be small (≤ 7)
  - commonly, there are an odd number of terms
- the terms should not overlap too much
- terms should overlap at around 0.5
- all terms are normal (max. membership is 1)
- all terms are convex
- the terms should span the universe of discourse

Note that these principles are simply guidelines for initial configurations of terms. If (as is probable) the membership functions of the terms are subject to tuning or optimisation procedures, then some (or many) of the principles may be violated.

There is little guidance that can be given as to the most appropriate shape of membership function for a given application domain. The problem of choosing an appropriate shape is exacerbated by the fact that there is no formal definition or interpretation as to the meaning of membership grades [23]. The main shapes used in practical applications are:

- piecewise linear (triangular or trapezoidal)
- Gaussian: $e^{\frac{-(x-c)^2}{\sigma^2}}$
- sigmoidal: of the form $\frac{1}{1+e^{-a(x-c)}}$
- double-sigmoidal: left and right sigmoidals put together to form a term with a central peak

where c, σ and a are parameters that determine the centre and spread of the Guassian or sigmoidal. Again, during development of the umbilical acid-base FES it was found both that clinicians preferred the appearance of non-piecewise linear membership functions (in fact, a combination of sigmoidals and double-sigmoidals were used) and that these membership functions gave a slight performance gain. Many authors, particularly in the context of fuzzy control, prefer to use piecewise linear functions, probably due to their ease of calculation, whereas other such as Mendel [26] prefer Gaussians because they are more simply analysable (e.g. differentiable) and hence are more suitable for automated tuning methodologies.

In the context of a fuzzy expert system, the source of membership functions would ideally be domain experts. However, again, there is no generally accepted method for eliciting membership functions from experts. An excellent survey of methods for membership function elicitation and learning is given by Krishnapuram [23]. The methods include:

- **Membership functions based on perceptions** — in which human subjects are polled or questioned.
- **Heuristic methods** — guessing!
- **Histogram-based methods** — in which membership functions are somehow matched to histograms of the base variables obtained from real data.
- **Transformation of probability distributions to possibility distributions** — if membership functions are considered numerically equivalent to possibility distributions and probability distributions are available, then methods are available to transform the probability distributions to possibility distributions.
- **Neural-network-based methods** — standard feedforward neural networks can be used to generate membership functions from labelled training data.
- **Clustering-based methods** — Clustering algorithms such as fuzzy c-means [4] can be used.

One further observation will be added. Most elicitation methods concentrate on direct or indirect elicitation of the membership functions themselves — i.e. they operate on the input side of the FES. During knowledge elicitation for the umbilical acid-base FES, it was found that such methods led to poor performance. That is, if an expert indicated membership functions directly, then the overall agreement of the FES with *that expert* was poor. In contrast, it was found that if the expert indicated desirable or undesirable features in the output consequent sets for specific (training) cases, then the input membership functions could be adjusted by a sort of *informal* back-propagation methodology to achieve better agreement with the expert.

4.1 Hints and Tips

- The number of linguistic variables is generally considered a component of model *structure*; the location and spread (and sometimes number) of membership functions are generally considered to be *parameters* of the model which should be tuned or optimised in some way.
- The choice of shape of membership function (m.f.) (e.g. piecewise linear, Gaussian or sigmoidal) is likely to be less important than the location of the centre of the m.f. and spread of the m.f.
- If a sizeable data set is available, then the use of automated m.f. learning methods should be considered.
- If data is difficult to obtain or human domain expertise is available, then manual knowledge elicitation methods might be favoured.

5 Rule Determination

Many of the comments that have just been made pertaining to linguistic variables and the membership functions of their terms also apply to rule determination. Indeed, in any practical construction of an FES the two processes will be closely interdependent. The linkage may be made more explicit by the strong statement that *membership functions have no meaning without the associated rules which will used to perform inference* — i.e. the rules provide the domain context necessary to give meaning to the linguistic variables and membership functions.

Once again, there is no single rule determination methodology that can be recommended. Some of the available options include:

- Elicitation of rules from experts. Although this is often presented as a self-evident option, in practice it is far from obvious how to undertake this. Unless only one expert is used for rule elicitation, there are bound to be disagreements, but how should such disagreements be overcome? This is currently an open question, which might be addressed by, for example, type-2 fuzzy logic methods [26].
- Conversion of rules from a non-fuzzy expert system. In the case of the umbilical acid-base FES, for example, the rules to construct a crisp expert system were initially elicited from experts. Once this crisp system had been successfully evaluated, it was converted to a fuzzy expert system. This allowed direct performance comparisons to be carried out, which confirmed that the fuzzy system indeed out-performed the crisp system [12].
- Crisp rule induction methods. If data is more plentiful than domain expertise then crisp rule induction methods such as Quinlan's ID3 or C4.5 algorithms [32, 33] can be utilised to produce a set of crisp rules that can consequently be fuzzified. Methods for doing this automatically have been proposed by Jang [17] and Baroglio [3].
- Evolutionary learning of rules (see Chaps. 8 & 9).
- Neuro-fuzzy approaches (see Chaps. 7 & 9) or, for example, the methods of Jang [16] and Wang and Mendel [41].

5.1 Hints and Tips

- Consider determination of the membership functions and the rules as two components of a single logical step in the design process.
- Whenever possible, use good sources of domain knowledge to lead or guide rule determination. However, always consider including rules generated from domain experts within tuning and optimisation processes.
- Purely automated rule determination procedures can lead to a large number of rules, and rules that may *conflict* (rules with the same antecedents but differing consequents). Most successful fuzzy expert systems feature considerably less than 100 rules. If use of an automated procedure for rule determination results in more than this, then some form of rule pruning [5, 28] should be considered.

6 Defuzzification

Once the fuzzy reasoning has been completed it is usually necessary to present the output of the reasoning in a human understandable form, through a process termed *defuzzification*. There are two principal classes of defuzzification, *arithmetic defuzzification* and *linguistic approximation*. In arithmetic defuzzification a mathematical method is used to extract the single value in the universe of discourse that 'best' (in some sense) represents the arbitrarily complex consequent fuzzy set (Fig. 5). This approach is typically used in areas of control engineering where some crisp result must be obtained. In linguistic approximation the primary terms in the consequent variable's term set are compared against the actual output set in a variety of combinations until the 'best' representation is obtained in natural language. This approach might be used in expert system advisory applications where human users view the output, although its actual usage has been relatively rare in the literature.

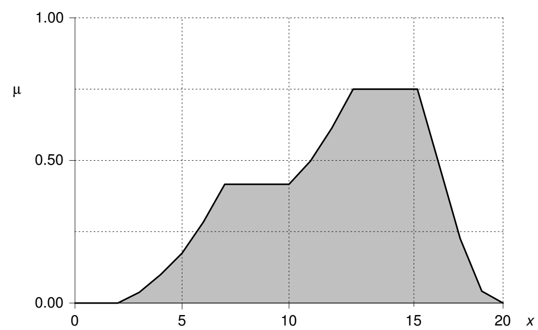

Fig. 5. A typical arbitrarily complex fuzzy output set

6.1 Arithmetic Defuzzification

The two most popular methods of arithmetic defuzzification are the *centre-of-gravity* (COG) algorithm and the *mean-of-maxima* algorithm. For the consequent set $A = \mu_1/x_1 + \mu_2/x_2 + \ldots + \mu_N/x_N$, the centre-of-gravity algorithm provides a single value by calculating the imaginary balance point of the shape of the membership:

$$x_g = \frac{\sum_{i=1}^{N}(\mu_i \cdot x_i)}{\sum_{i=1}^{N}\mu_i} \tag{4}$$

The mean-of-maxima algorithm finds the point in the universe of discourse with maximum membership grade:

$$x_m = \max_i \mu_i \tag{5}$$

and calculates the mean of all the maxima if more than one maximum is found. An illustration of the output of these two methods is shown in Fig. 6. Unfortunately, both these methods have problems. Firstly, they both obviously lose information by trying to represent a complex fuzzy shape as a single scalar number. The COG method is insensitive to the overall height of the fuzzy consequent set, and the mean-of-maxima is prone to discontinuities in output, as only a small change in shape (for instance if there are two similar sized peaks) can cause a sudden large change in output value.

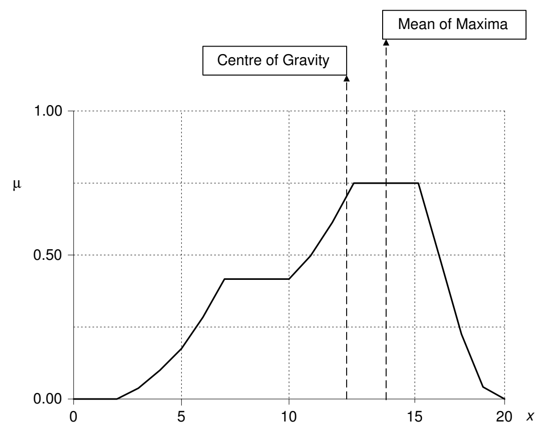

Fig. 6. An illustration of the difference between *centre-of-gravity* and *mean-of-maxima* defuzzification

A number of alternative parameters can also be calculated to provide more information on the shape of the output as well as its location. A variety of such parameters are described, and illustrated through the example fuzzy output sets, A, B, C, D, E, and F, as shown in Fig. 7.

Membership Grade at the Defuzzification Point

The membership grade of the output set at the centre-of-gravity, μ_g, or mean-of-maxima, μ_m, provides an indication of confidence in the result. For example, in Fig. 7, the COG (x_g) of set D is 50, and the membership value at this point (μ_g) is 0.90. Actually, in the same Figure, the output fuzzy sets D and E have the same COG ($x_g = 50$), but set D has a higher confidence in the result ($\mu_g = 0.90$) than set E ($\mu_g \approx 0.16$).

Maximum Membership Grade

The maximum membership grade attained by the consequence fuzzy set, μ_{max}, provides a direct measure of the maximum strength that the an-

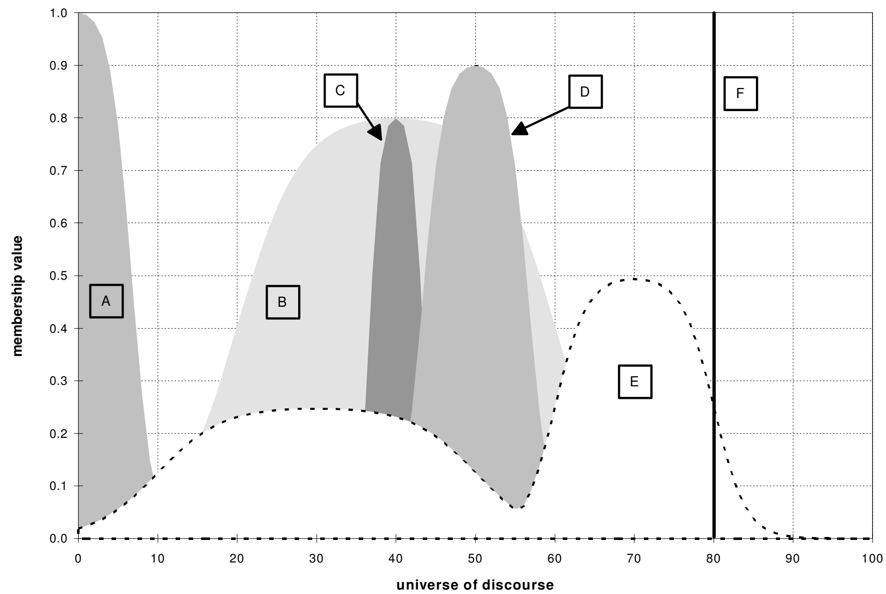

Fig. 7. Illustration of a variety of fuzzy output sets, A, B, C, D, E, and F, with different defuzzification parameters

tecedents fired a rule. It is especially useful for edge sets such as set A in Fig. 7, as the centroid cannot be at the maximum point, so that $\mu_g < \mu max$.

Normalised Area

The area of the output set normalised to its maximum value is given by:

$$area = \frac{\sum_{i=1}^{N} \mu_i}{N} \tag{6}$$

This gives a value of 1.0 for the *unknown* set ($\mu = 1$ across the universe of discourse), a value of 0.0 for the *undefined* set ($\mu = 0$ across the universe of discourse), and would give a minimal value (≈ 0) for a fuzzy singleton. In Fig. 7 the output fuzzy sets B and C have the same COG ($x_g = 40$) and the same membership at this point ($\mu_g = 0.80$), but set B has a larger area and hence a larger uncertainty.

Fuzzy Entropy

Yet another measure, termed the *entropy* of a fuzzy set is defined by:

$$S = \frac{\sum_{i=1}^{N}(-\mu_i \log_2(\mu_i) - (1-\mu_i)\log_2(1-\mu_i))}{N} \tag{7}$$

This is normalised to its maximum value to give a value between zero and one, and provides an indication of the lack of information contained in the output set in terms of the distance away from the extremes of $\mu = 0.0$ and $\mu = 1.0$. It therefore gives a value of 0.0 for the *unknown* and *undefined* sets, and gives a value of 1.0 for the *indeterminate* set ($\mu = 0.5$ across the universe of discourse). Similarly to the normalised area, it too gives a minimal value for fuzzy singletons.

Summary of Arithmetic Defuzzification

Table 1 summarises the various arithmetic defuzzification parameters defined above for sets A, B, C, D, E, and F, in Fig. 7, and for the level sets *unknown*, *indeterminate* and *undefined*. It can be seen that for the fuzzy singleton F, which represents a crisp output from a fuzzy system, the values for μ_g and μ_{max} are both high (1.00) and the values for *area* and *entropy* are both low (0.00). The same tendencies can be seen for the example sets A, C, and D (only). There are a whole range of possibilities for other measures, such as the span of the fuzzy set (percentage of the set that is non-zero), the percentage of total area within a certain distance of the centroid point, the number of peaks in the fuzzy set, and so on.

Table 1. Summary of various arithmetic defuzzification parameters for sets A, B, C, D, E, and F, in Fig. 7, with values obtained for level sets *unknown*, *indeterminate* and *undefined*

Set	x_g	μ_g	μmax	*area*	*entropy*
A	3	0.95	1.00	0.07	0.06
B	40	0.80	0.80	0.32	0.50
C	40	0.80	0.80	0.05	0.08
D	50	0.90	0.90	0.12	0.15
E	50	0.16	0.50	0.19	0.60
F (fuzzy singleton)	80	1.00	1.00	0.00	0.00
unknown (1.0/x)	50	1.00	1.00	1.00	0.00
indeterminate (0.5/x)	50	0.50	0.50	0.50	1.00
undefined (0.0/x)	50	0.00	0.00	0.00	0.00

6.2 Linguistic Approximation

In linguistic approximation a similarity measure is used to compute the distance between the actual output set and an arbitrary collection of primary terms, connectives and hedges. For example, a shower control variable with primitive terms such as *cold*, *medium* and *hot*, and allowable hedges of *fairly* and *very*, might produce a composite linguistic term such as *medium and* *fairly hot*. One such similarity metric is the Euclidean distance between fuzzy sets, given by:

$$\delta = \sqrt{\sum_i (\mu_i - \eta_i)^2} \tag{8}$$

where μ_i is the membership of the output set and η_i is the membership grade of the currently considered linguistic approximation — the minimum value of δ will determine the best match. Alternatively, the degree of overlap, γ, of two fuzzy sets, A and B, can be calculated by dividing the area of intersection by the area of the union of the sets:

$$\gamma = \frac{A \cap B}{A \cup B} \tag{9}$$

to give a value between zero (for disparate sets) and one (for coincidental sets) — the maximum value of γ will determine the best match.

A search is then initiated to find the best match whilst attempting to limit the complexity of the combination of terms, in order to produce comprehensible output. For example, although the linguistic combination *not extremely cold and fairly medium and medium and fairly hot* might produce a better match than *medium and fairly hot* for Fig. 6, the latter term would be preferred due to its relative simplicity.

6.3 Hints and Tips

- Defuzzification is a fundamentally important step in Mamdani inference. Although many alternatives for numeric defuzzification have been proposed, there is little theoretical guidance as to which is better in any given situation. It is probably safest to stick to centre-of-gravity defuzzification unless there is a good reason otherwise.
- Linguistic defuzzification has been infrequently used, and is a poorly understood and little researched area.
- Type-2 fuzzy logic [26] might be considered if linguistic output is required.

7 Evaluation

Many authors have used the terms *verification*, *validation*, *assessment* and *evaluation* in differing and inconsistent manners in the literature [29, 30]. In this section the following terminology, designed specifically for the European Advanced Informatics in Medicine (AIM) project [10], is adopted:

- *verification* is the process of ensuring that the expert system is functioning according to its specification,
- *validation* is the process of ensuring that the knowledge embedded within the expert system is an accurate representation of the domain, and

- *assessment* is the process of determining the effect that the expert system has in the real-world setting — this can be further split into two further sub-tasks:
  1. *human factors assessment* - determining whether the system is useful to and usable by its target users, and
  2. *performance assessment* - determining whether the system makes a measurable difference (improvement) when deployed.

Evaluation is a global term that refers to the collective processes of *verification*, *validation* and *assessment*.

In an ideal framework of scientific investigation the entire evaluation methodology would be established and fixed *a priori*. That is to say, the data, experiments, statistical tests and acceptable performance measures to be used in deciding whether an expert system was acceptable would all be decided before construction of the system commenced. This is very rarely the case in practical expert system development and fuzzy expert systems are no exception. It is not possible to cover the process of system evaluation thoroughly here; the reader is referred to [29, 30] for general guidance.

7.1 Model v. System Evaluation

As mentioned in the introduction there is a subtle distinction between evaluation of the theoretical fuzzy model (*model evaluation*) and evaluation of a specific implementation of a fuzzy expert system (*system evaluation*). In order to carry out a thorough system evaluation, it is necessary to verify and validate the implementation itself, in terms of examining the code to ensure that there are no coding bugs, examining the environment to ensure, for example, that variables cannot overflow, etc. In short, software engineering evaluation [31, 36] must be carried out in order to claim that an FES has been evaluated. On the other hand, fuzzy model evaluation can be carried out by tasks such as comparison of membership function shape / location with expert opinion, or discussion of rules-sets with experts, etc.

7.2 Data Categories

Data can be distinguished into three categories:

- *training data* — training data is used in the development of the model as part of the tuning or optimisation process;
- *testing data* — testing data is used to measure the performance of the system within model development process;
- *evaluation data* — evaluation data should be novel or entirely unused data that is used to evaluate performance after the final model has been fixed.

Often training data and testing data are simply logical partitions of one data set. For example, in two-fold cross validation the data is split into two halves.

Firstly, one half is used for training the system and when training has been completed the system is fixed and other half of the data is used for testing. The process is then repeated with the data sets swapped and the performance obtained from the two runs is pooled. Another common training and testing procedure is the 'leave-one-out' method, in which all the data except one instance are used for training the system which is then tested on the one remaining datum. This process is repeated, leaving a differing datum out, until the whole data set has been used once for testing. The performance results are then aggregated in some way (usually simply averaged).

Evaluation data should be an entirely separate data set which has not been used in any way in the construction, training (optimisation) or testing of the system. Indeed, it is scientifically desirable for the system designer to not have had any contact with the evaluation data at all, as any viewing or other appreciation of the data could unwittingly influence construction of the system. In the ideal situation, collection of novel evaluation data takes place *after* completion of system construction, but to pre-specified evaluation procedures. This rarely happens in practice.

7.3 Performance Measures

In order to evaluate a system, a measure of performance is necessary. Sometimes appropriates measures are obvious. As an example, take the financial forecasting FES. As the system is to advise when to trade shares, the obvious performance measure is the profit (or loss!) made when trading shares using real share prices. Either past data that has not been used for system training and testing could be used for evaluation or, better, real data collected after the system can be proven to have been fixed. Often some variation of mean-squared-error (MSE) between actual output and ideal (or desired) output is utilised.

However, frequently in expert systems (particularly medical expert systems) there is no objective measure of correct performance and other means must be used. The solution in such cases is usually to compare the output of the (fuzzy) expert system against human expert opinion on a range of data. An indirect measure of performance is then created. If the expert opinion and expert system output is categorical (e.g. classification into one of a number of specified disease categories) then the proportion of correct matches can be used. It is better to use a statistical method to correct for chance agreement such as the Kappa statistic [6], which can be used to measure either exact agreements only or can be modified to allow for partial agreement(s) [7]. In the case of the umbilical acid-base FES, a set of 50 difficult cases (see below) was selected for evaluation. Clinical experts and the fuzzy expert system were then asked to rank those cases from worst to best in terms of their indication of the infant's state of health. Spearman rank order correlation, effectively a form of MSE, was then used to measure agreement.

Imagine, as often the case, that the result of an FES is a number on a continuous scale obtained by, for example, centre-of-gravity defuzzification. If this number is representing, say, diagnosis of the presence of a disease, then what threshold should be used to indicate that the disease is indeed present? Often an arbitrary threshold is chosen (e.g. 50 on a scale $0 \ldots 100$). However, a better solution is to use a technique known as Receiver Operating Characteristic (ROC) curves [9, 38] in which the threshold is continuously varied in order to achieve the best agreement. Note that, if ROC analysis is used, a method of generating accurate confidence intervals should be employed [40].

Picking suitable data to be used for evaluation purposes is another tricky area. Ideally, suitable data should cover the range of output possibilities of the fuzzy expert system in a systematic manner. But how can this be known until after evaluation has been carried out? There is no simple answer to this. Probably the best solution, if possible, is to have an independent, acknowledged domain expert pick a suitable set of cases. However, it is often difficult to find such an expert and undesirable, as is technically necessary, to then not use this expert for the evaluation exercise.

7.4 Hints and Tips

- Consider the method of evaluation at the earliest possible stage of system construction. Give serious thought to how rigorous and reliable evaluation will be assured.
- Consider incorporating a tunable performance parameter into the final model and use ROC analysis or similar to evaluate the system.

8 Model Optimisation

Model optimisation will be covered in several Chapters of this book, so details will not be duplicated here. Some points of specific relevance to fuzzy expert system optimisation will be made. Firstly, some form of optimisation is an essential part of developing any fuzzy expert system. It may be used to adapt a fuzzy expert system in a certain domain to changes in the environment, i.e. adaption in the first category. Effectively, it can be viewed as the process of adapting an FES to the specific characteristics of the training and testing data available at the time. Thus the optimisation process could be applied at several stages in the lifetime of an FES in order to update the knowledge embodied in the FES to respond to drifts in the environment. However, in current state-of-the-art such an adapted FES would have to be (manually) re-evaluated. Accepted practice for automatically adapting a fuzzy expert system and evaluating it have not yet been developed.

Secondly, any optimisation technique may be used. Probably the most popular method currently is some form of evolutionary optimisation (see Chap.

13), but other techniques have been used (see below). Thirdly, before undertaking (for example) complex evolutionary optimisation, it is probably worth attempting more simplistic methods such as sampling, hill climbing or Monte Carlo approaches. Using any method, it is unlikely that a true global optimum of performance will be determined. It is much more likely that optimisation will be halted when some predetermined or subjective threshold of performance has been reached, or a time limit has been passed. It might well be the case that implementation of a simple Monte Carlo optimisation will allow adequate performance to be achieved in a similar time taken to program complex evolutionary optimisation.

Finally, the most significant influences on the performance of an FES are likely to be the number and location of the membership functions and the number and form of the rules. The precise shape of membership function (as in triangular, Gaussian, double-sigmoidal, etc., see Sect. 4) is unlikely to make a sizeable difference. Finally, *do not forget data preparation*!

Available FES optimisation techniques include:

- Exhaustive Search — The practice of systematically altering all parameters by the smallest interval such that *entire* coverage of the entire search space is obtained. This is unlikely to be feasible for an expert system with realistic dimensions.
- Sampling — The practice of systematically altering parameters by fixed, discrete intervals of sufficient size such that a *reasonable* coverage of the entire search space is obtained. The difference between this and exhaustive search can be clarified with an example. Suppose a search is carried out to locate the optimal value for the centre of a triangular membership function across an integer universe of discourse from 0 to 100. In exhaustive search, all 101 values (0 to 100) would be considered. In sampling, perhaps only 9 values (e.g. 10, 20, ..., 90) would be considered. Clearly, the optimal value may not be found in this way.
- Hill Climbing — The practice of systematically altering parameters (sometimes independently) in direction of greatest improvement until no further improvement is obtained.
- Monte Carlo — The process of assigning parameter values randomly until an acceptable solution (parameter set) is found. Monte Carlo approaches can be combined with Hill Climbing.
- Simulated Annealing — For a detailed review of Simulated Annealing, see Aarts [1]. Simulated Annealing is most often applied to combinatorial optimisation, but it can also be applied to continuous optimisation of both the structure and parameters of fuzzy models [11].
- Evolutionary Algorithms — See Chaps. 7, 9 & 13 for further details and references.
- Neuro-fuzzy or other hybrid approaches — See Chaps. 7 & 15 for further details and references.

- Other — Other generic optimisation techniques such as, for example, ant colony optimisation might be applied, but little research has been carried out in such areas.

Although different methods are preferred by different authors, there is no general result to state that any one optimisation technique is guaranteed to be better than any other. Indeed, the No Free Lunch Theorem [42] states that all optimisation techniques perform *on average* the same over all optimisation problems. This does not necessarily apply to any given subset of optimisation problems (e.g. specifically to optimisation of fuzzy models) and it does not mean that if one is experienced at applying a particular technique (e.g. evolutionary optimisation) then one should not continue to use that technique. However, in the absence of contrary evidence, it does suggest that there is no theoretical reason to favour e.g. evolutionary optimisation over e.g. simulated annealing optimisation. From a practical point of view, there are strong arguments in favour of using evolutionary algorithms. They are relatively easily applied, general purpose techniques that are well suited to the multi-modal, discontinuous and non-differentiable nature of the 'landscape' of the evaluation functions often found.

9 Authoring Tools

There are a number of commercially available and freely available authoring tools for creating fuzzy expert systems. The most comprehensive (but, unfortunately, possibly the most expensive) is MATLAB® together with the optional Fuzzy Logic Toolbox. This is a comprehensive package that provides Mamdani and TSK inference, together with many choices of membership function generation and fuzzy operators, all accessible via a command line interface or an advanced graphical user interface with graphical capabilities for constructing and evaluating the output of fuzzy expert systems. MATLAB® also provides a comprehensive programming environment that allows data preprocessing and optimisation techniques to be coded and automated. Other fuzzy methodologies such as ANFIS (Adaptive-Network-Based Fuzzy Inference System) [16] and Fuzzy C-Means clustering [4] are also provided.

Freely available tools include Fuzzy CLIPS. To quote from the Fuzzy CLIPS website [46]:

> FuzzyCLIPS is an extension of the CLIPS (C Language Integrated Production System) expert system shell from NASA. It was developed by the Integrated Reasoning Group of the Institute for Information Technology of the National Research Council of Canada and has been widely distributed for a number of years. It enhances CLIPS by providing a fuzzy reasoning capability that is fully integrated with CLIPS facts and inference engine allowing one to represent and manipulate

> fuzzy facts and rules. FuzzyCLIPS can deal with exact, fuzzy (or inexact), and combined reasoning, allowing fuzzy and normal terms to be freely mixed in the rules and facts of an expert system. The system uses two basic inexact concepts, fuzziness and uncertainty. It has provided a useful environment for developing fuzzy applications but it does require significant effort to update and maintain as new versions of CLIPS are released.

Note that FuzzyCLIPS is free for educational and research purposes only; commercial uses of the software **require** a commercial licence. Full licence details are available from the FuzzyCLIPS website. Several books such as Cox [8] also supply various fuzzy toolsets, although the capabilities, flexibility and quality of the software varies. Of course, it is also relatively straightforward to code a fuzzy expert system from 'first principles' using any programming language of choice, including low-level languages such as C, C++, Java or higher-level languages such as Python [47].

10 Future Directions

Leaving aside difficulties in defining 'smart', it is probably fair to state that, strictly, classical (non-hybrid) fuzzy expert systems are currently neither smart nor adaptive. While this may sound like a harsh assessment, the reality of the current position is that fuzzy methods that are adaptive and are much closer to being 'smart' are to be found in neuro-fuzzy, evolutionary fuzzy or other hybrid methods as covered in Chaps. 7, 8 & 9. It is in these hybrid methods that much leading-edge research is currently focussed.

So, what are the future directions for classical (non-hybrid) fuzzy expert systems? This is a difficult question to address with any clarity or certainty. Zadeh continues to attempt to promote a move to 'computing with words' [44]; a paradigm that has yet to be realised. Partly to address this, Zadeh has recently introduced the concept of a protoform [45], the implementation of which Kacprzyk has investigated (See Chap. 16). Other current research into computing with words is examining type-2 fuzzy logic, in which membership functions are 'blurred' to represent uncertainty in the membership values associated with any specific value of the base variable. Although first introduced by Zadeh in 1975 [43], type-2 fuzzy logic was largely neglected until particularly Mendel and his group reactivated interest more recently. Research activity and hence publications in the area are now proliferating, for example [13, 18, 19, 20, 25]. For introductory material on type-2 fuzzy logic see Mendel's recent book [26] or Mendel and John's tutorial paper [27].

11 Conclusions

The construction of a successful fuzzy expert system is a difficult process that requires considerable expertise from the fuzzy knowledge engineer. It is probably the large number of design choices and tunable parameters of the resultant expert system that are partly responsible for the success of fuzzy expert systems. It is not currently possible to automate the entire process so that a 'generic' fuzzy expert system may be designed which will then automatically adapt itself somehow to each new environment it encounters, although frameworks such as Abraham's EvoNF [2] or systems such as Jang's ANFIS [16] can automate many of the steps. Fully automated 'smart' adaptation is very much an ongoing research issue.

References

1. E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* John Wiley & Sons, New York, 1989.
2. A. Abraham. EvoNF: A framework for optimisation of fuzzy inference systems using neural network learning and evolutionary computation. In *Proceedings of the 17th IEEE International Symposium on Intelligent Control (ISIC'02)*, pages 327–332, IEEE Press, 2002.
3. C. Baroglio, A. Giordana, M. Kaiser, M. Nuttin, and R. Piola. Learning control for industrial robots. *Machine Learning*, 2/3:221–250, 1996.
4. J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum, New York, 1981.
5. J.L. Castro, J.J. Castro-Schez, and J.M. Zurita. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems*, 101:331–342, 1999.
6. J. Cohen. A coefficient of agreement for nominal scales. *Educational Psychological Measurement*, 20:37–46, 1960.
7. J. Cohen. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220, 1968.
8. E. Cox. *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using and Maintaining Fuzzy Systems.* AP Professional, San Diego, CA, second edition, 1998.
9. J.P. Egan. *Signal Detection Theory and ROC Analysis.* Academic Press, New York, 1975.
10. R. Engelbrecht, A. Rector, and W. Moser. Verification and validation. In E.M.S.J. van Gennip and J.L. Talmon, editors, *Assessment and Evaluation of Information Technologies*, pages 51–66. IOS Press, 1995.
11. J.M. Garibaldi and E.C. Ifeachor. Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation. *IEEE Transactions on Fuzzy Systems*, 7(1):72–84, 1999.
12. J.M. Garibaldi and E.C. Ifeachor. The development of a fuzzy expert system for the analysis of umbilical cord blood. In P. Szczepaniak, P.J.G. Lisboa, and J. Kacprzyk, editors, *Fuzzy Systems in Medicine*, pages 652–668. Springer-Verlag, 2000.

13. J.M. Garibaldi and R.I. John. Choosing membership functions of linguistic terms. In *Proceedings of the 2003 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2003)*, pages 578–583, St. Louis, USA, 2003. IEEE, New York.
14. J.M. Garibaldi, J.A. Westgate, and E.C. Ifeachor. The evaluation of an expert system for the analysis of umbilical cord blood. *Artificial Intelligence in Medicine*, 17(2):109–130, 1999.
15. J.M. Garibaldi, J.A. Westgate, E.C. Ifeachor, and K.R. Greene. The development and implementation of an expert system for the analysis of umbilical cord blood. *Artificial Intelligence in Medicine*, 10(2):129–144, 1997.
16. J.-S.R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.
17. J.-S.R. Jang. Structure determination in fuzzy modeling: A fuzzy CART approach. In *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94)*, pages 480–485, Orlando, FL, 1994.
18. R.I. John. Embedded interval valued fuzzy sets. *Proceedings of FUZZ-IEEE 2002*, pages 1316–1321, 2002.
19. R.I. John, P.R. Innocent, and M.R. Barnes. Neuro-fuzzy clustering of radiographic tibia image data using type-2 fuzzy sets. *Information Sciences*, 125/1-4:203–220, 2000.
20. N.N. Karnik and J.M. Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems*, 122:327–348, 2001.
21. N.K. Kasabov. *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. MIT Press, Cambridge, Massachussets, 1996.
22. G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice-Hall, Upper Saddle River, NJ, 1995.
23. R. Krishnapuram. Membership function elicitation and learning. In E.H. Ruspini, P.P. Bonissone, and W. Pedrycz, editors, *Handbook of Fuzzy Computation*, pages B3.2:1–B3.2:11. Institute of Physics, Bristol, UK, 1998.
24. E.H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7:1–13, 1975.
25. J.M. Mendel. On the importance of interval sets in type-2 fuzzy logic systems. In *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pages 1647–1652, Vancouver, Canada, 2001. IEEE, New York.
26. J.M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall PTR, 2001.
27. J.M. Mendel and John R.I. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, 2002.
28. D. Nauck and R. Kruse. A neuro-fuzzy approach to obtain interpretable fuzzy systems for function approximation. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'98)*, pages 1106–1111, Anchorage, AK, 1998.
29. R.M. O'Keefe, O. Balci, and E.P. Smith. Validating expert system performance. *IEEE Expert*, 2(4):81–90, 1987.
30. T.J. O'Leary, M. Goul, K.E. Moffitt, and A.E. Radwan. Validating expert systems. *IEEE Expert*, 5(3):51–58, 1990.
31. R.S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw Hill, Maidenhead, UK, fifth edition, 2000.

32. J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
33. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
34. E.H. Ruspini, P.P. Bonissone, and W. Pedrycz, editors. *Handbook of Fuzzy Computation*. Institute of Physics, Bristol, UK, 1998.
35. O. Siggaard-Andersen. *The Acid-Base Status of the Blood (4th edn)*. Munksgaard, Copenhagen, 1976.
36. I. Sommerville. *Software Engineering*. Addison-Wesley, Harlow, UK, sixth edition, 2001.
37. M. Sugeno and G.T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:15–33, 1988.
38. J.A. Swets and R.M. Pickett. *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*. Academic Press, New York, 1982.
39. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
40. J.B. Tilbury, P.W.J. Van-Eetvelt, J.M. Garibaldi, J.S.H. Curnow, and E.C. Ifeachor. Reciever operator characteristic analysis for intelligent medical systems - a new approach for finding confidence intervals. *IEEE Transactions on Biomedical Engineering*, 47(7):952–963, 2000.
41. L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, 1992.
42. D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.
43. L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - I,II,III. *Information Sciences*, 8;8;9:199–249;301–357;43–80, 1975.
44. L.A. Zadeh. From computing with numbers to computing with words — from manipulation of measurements to manipulation of perceptions. *IEEE Transactions on Circuits and Systems*, 45(1):105–119, 1999.
45. L.A. Zadeh. A prototype-centered approach to adding deduction capability to search engines — the concept of protoform. *Proceedings of NAFIPS 2002*, pages 523–525, 2002.
46. National Research Council Canada, Institute for Information Technology, Fuzzy CLIPS Website, `http://www.iit.nrc.ca/IR_public/fuzzy`.
47. Python Website, `http://www.python.org`.