Decision Support Systems 31 (2001) 263–285

Decision Support Systems

www.elsevier.com/locate/dsw

Using soft computing to build real world intelligent decision support systems in uncertain domains

John Zeleznikow [a,*], James R. Nolan [b]

[a] Department of Computer Science, Computer Engineering, La Trobe University, Bundoora, Victoria 3083, Australia
[b] Departments of Quantitative Business and Computer Science, Siena College, 515 Loudon Road, Loudonville, NY, USA

Abstract

Whilst the builders of traditional decision support systems have regularly used game theory and operations research, they have rarely used statistical techniques to build intelligent support systems for fields that have weak domain models. Further, the principle tools in the artificial intelligence arsenal were centred on symbol manipulation and predicate logic, while the use of numerical techniques were looked upon with disfavour.

We claim that soft computing techniques (such as fuzzy reasoning and neural networks) can be integrated with symbolic techniques to provide for efficient decision making in knowledge-based systems. We illustrate our claim through the discussion of two decision support systems that have been constructed using soft computing techniques. Split-Up uses rules and neural networks to advise on property distribution following divorce in Australia, whilst IFDSSEA uses fuzzy reasoning to assists teachers in New York State to grade essays.

We focus on how both systems reason and how they have been evaluated.

Keywords: Soft computing; Knowledge discovery; Fuzzy reasoning; Neural networks; Evaluation of decision support systems

1. An introduction to decision support systems and soft computing

According to Turban and Aronson [32] a decision support system (DSS) is a computer-based information system that combines models and data in an attempt to solve non-structured problems with extensive user involvement. They claim an expert system (ES) is a computer system that applies reasoning methodologies on knowledge to render advice or recommendations much like a human expert. When expert systems technology was first applied to decision-making problems, it fell short in several respects.

Early expert systems were rule-based. They were not capable of handling the classical DSS functions that are more computational than logical. Recently, artificial intelligence researchers have seen the necessity of using statistical techniques to build intelligent decision support systems [17,33]. Examples of such statistical techniques include fuzzy logic, neural networks, rule induction and various Bayesian techniques.

Turban and Aronson [32] claim that although uncertainty is widespread in the real world its treatment in the practical world of artificial intelligence is

---

[*] Corresponding author.

E-mail addresses: johnz@cs.latrobe.edu.au (J. Zeleznikow), JNolan@siena.edu (J.R. Nolan).

PII: S0167-9236(00)00135-4

very limited. In this paper we consider in detail soft computing techniques for building intelligent decision support systems in the presence of uncertainty.

1.1. Soft computing

Zadeh [37,38] claims that the most important factor in soft computing is the potential to mimic the ability of the human mind to effectively employ modes of reasoning that are approximate rather than exact. He states that in traditional-hard-computing, the prime desiderata are precision, certainty and rigour. By contrast, the point of departure in soft computing is the thesis that precision and certainty carry a cost and that computation, reasoning, and decision making should exploit—wherever possible—the tolerance for imprecision and uncertainty. In raising the banner of '*Exploit the tolerance for imprecision and uncertainty*' soft computing uses the human mind as a role model and, at the same time, aims at a formalisation of the cognitive processes humans employ so effectively in the performance of daily tasks. Zadeh states that the principal constituents of soft computing are (i) fuzzy logic, (ii) neural network theory and (iii) probabilistic reasoning with the latter subsuming belief networks, genetic algorithms, parts of learning theory and chaotic systems.

Zeleznikow and Hunter [40] discuss how to provide for reasoning with uncertainty in decision support systems. Techniques discussed include (a) fuzzy logic, (b) neural networks, (c) Bayesian inference, (d) Dempster–Shafer theory of belief functions and (e) certainty factors. In this paper we shall focus upon using fuzzy logic and neural networks for building intelligent decision support systems in the presence of uncertainty.

Fuzzy logic is a many valued propositional logic where each proposition P rather than taking the value true or false has a probability attached of being true. Logical operators and probability theory are then combined to model reasoning with uncertainty.

A neural network receives its name from the fact that it resembles a nervous system in the brain. It consists of many self-adjusting processing elements cooperating in a densely interconnected network. Each processing element generates a single output signal that is transmitted to the other processing elements. The output signal of a processing element depends on the inputs to the processing element: each input is gated by a weighting factor that determines the amount of influence that the input will have on the output. The strength of the weighting factors is adjusted autonomously by the processing element as data is processed.

Bayesian methods provide formalism for reasoning about partial beliefs under conditions of uncertainty. In this formalism, propositions are given numerical values, signifying the degree of belief accorded to them. Dempster–Shafer theory has been developed to handle partially specified domains. It distinguishes between uncertainty by creating belief functions. Belief functions allow the user to bound the assignment of probabilities to certain events, rather than give events specific probabilities. A certainty factor in an expert system is a probability that the conclusion reached by the system is correct.

In Zadeh's classification, fuzzy logic is primarily concerned with imprecision, neural networks with learning and probabilistic reasoning with uncertainty. It is important to notice that there are areas of overlap between the techniques.

Nolan [17] reports on the development and implementation of DISEXPERT, an intelligent rule-based system tool for the referral of United States social security disability recipients to vocational rehabilitation services. The system provides support to paraprofessional case workers in reaching unbiased and consistent assessment decisions regarding referral of clients to vocational rehabilitation services. An artificial intelligence approach was used since successful intelligent systems have been built in risk assessment domains [12].

Domain experts identified 22 factors as important for predicting successful vocational rehabilitation. Each of the expert vocational rehabilitation counselors was given the same 225 disability cases to review and make an assessment. A rule induction system 1ST-CLASS FUSION (which uses the inductive learning algorithm ID3) was utilised to produce rules made up of relationships among the previously identified statistically relevant factors.

The results after 4 years of use of DISEXPERT, demonstrate that paraprofessionals using DISEXPERT can make assessments in less time and with a level of accuracy superior to the vocational rehabilitation domain professionals using manual methods.

Proper testing and validation of any system is important for determining the accuracy, completeness and performance of a system [18]. As part of the evaluation of DISEXPERT, the system was tried on 200 cases that were reviewed by domain experts. DISEXPERT agreed with domain experts on 198 of the 200 cases.

Bellucci and Zeleznikow [4] have conducted research into employing the use of paradigms to support cognitive support processes. Cognitive mapping has previously been used to build negotiation support systems in the domain of international negotiation [7]. In these applications, cognitive mappings were used to examine concepts as well as the influence of background knowledge and to extract explanations for consequences. A fuzzy cognitive map resembles a neural network in its structure, but uses fuzzy variables instead of real numbers to represent nodes. The directed connections between nodes represent a causal relationship between two nodes. Kosko [15] has defined a fuzzy cognitive map ‘as modelling dynamic concepts or actors which are interrelated in complex ways’, which concurs with our view of negotiation. Fuzzy cognitive mapping is a paradigm used in domains where numerical data is uncertain or hard to retrieve. Kosko [15] writes ‘They (fuzzy cognitive maps) help most in the value clashes that mix head and heart’. The application of [4] relates to Australian family law, a domain in which little numerical data is available, and conflict is usually resolved amidst highly emotional disputes. They hence justify the use of cognitive mapping as an appropriate paradigm to model the family law negotiation domain.

1.2. Reasoning with uncertainty and imprecision

As Nolan [16] states, intelligent decision support systems must have the ability to process both quantitative and qualitative data (at varying levels of precision) and use reasoning to transform data into opinions, judgments, evaluations and advice. Intelligent decision support systems must be able to exploit a tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness, low solution cost and a better rapport with reality. This is especially true when building real world applications. Traditional artificial intelligence uses deductive reasoning. Soft computing, whether it be fuzzy logic, neural network theory or probabilistic reasoning requires reasoning with imprecision and the integration of statistical techniques with traditional symbolic reasoning. Because of the use of statistical techniques, soft computing decision support systems require a separate explanation facility to indicate why the statistically derived answer is valid. In this article we discuss how to provide explanation for such systems, and how to evaluate both the performance of the decision making and explanation processes in such systems.

We also consider how soft computing techniques (such as fuzzy logic and neural networks) can be used to address some of the problems mentioned above. We illustrate the significance of the issues raised in this paper through the discussion of two real-world intelligent decision support systems we have built. It is important to stress that the two systems were independently designed and implemented. We do, however, believe that much valuable knowledge can be obtained by a comparison of the two systems.

The first system (Split-Up) advises on property distribution following divorce in Australia, whilst the second system (IFDSSEA) assists teachers in New York State to grade essays. A detailed discussion of the manner in which Split-Up reasons can be found in Ref. [28] whilst similar information about IFDSSEA can be found in Ref. [16]. Both systems make heavy use of domain expertise and knowledge engineering, as did DISEXPERT described above. They are both hybrid systems using a combination of deductive and soft computing techniques.

A hybrid reasoning system combines facets of one or more representation schemes into a single integrated programming environment. It usually includes object orientation, rules for representing heuristic knowledge and support for a variety of search strategies. At the Donald Berman Laboratory for Information Technology and Law we have used hybrid reasoning in the manner described in Table 1.

IFDSSEA has been constructed and evaluated in a manner similar to DISEXPERT. Split-Up has been more rigorously evaluated, as shall be seen from our discussion of the issue. IFDSSEA is used by teachers in New York State whilst Split-Up is being tried by Australian judges, registrars mediators and lawyers.

Table 1

System	Application	Reasoning techniques used
IKBALS (Zeleznikow [39])	Workers compensation	Rule-based reasoning and case-based reasoning
IKBALS III (Zeleznikow et al. [44])	Credit law	Rule-based reasoning and case-based reasoning. Rule induction was used to learn factors about closest factors
Split Up (Stranieri et al. [28])	Family law, Property distribution	Rule-based reasoning and neural networks. Explanations are provided by a separate (Toulmin) Argument Shell.
Family_Negotiator (Bellucci and Zeleznikow [2])	Family law, Negotiation	Rule-based reasoning and case-based reasoning
Embrace (Yearwood and Stranieri [36])	Refugee law	Rule-based reasoning and information retrieval
Family_Winner (Bellucci and Zeleznikow [4])	Family law, Negotiation	Rule-based reasoning, case-based reasoning and fuzzy cognitive maps

Following further improvements (discussed in the paper) and evaluation, Split-Up will be marketed commercially.

2. Reasoning in discretionary domains

Black [6] claims that discretion is a power or right conferred upon decision-makers to act according to the dictates of their own judgment and conscience, uncontrolled by the judgment or conscience of others. Nevertheless, decision-makers must in accordance with the rule of law and their decisions must preserve the rights of all parties effected by the decision-making. It is thus essential that discretionary decision making not be arbitrary—since an arbitrary application of discretion could lead to enhanced conflict by any aggrieved parties. Rather than appear to make random decisions, they wish to develop a measure of consistency to their decision making. But how do we model discretionary domains?

Stranieri et al. [28] concluded that the important features for modeling legal domains are the extent to which a task is both open textured and bounded. Open textured predicates contain questions that cannot be structured in the form of production rules or logical propositions and which require some domain knowledge on the part of the user in order to answer. We consider a well-defined predicate to be the opposite of an open-textured predicate.

An example of a well-defined domain is that of the punishment of drunk drivers. In Australia there is legislation which says ‘If you drive while drunk then you will lose your licence.’ This can be written as a rule:

$$\text{drink and drive} \rightarrow \text{licence\_loss};$$

where: drink means having a level of alcohol in one’s blood above a certain limit (in Victoria this is 0.05%); drive means driving a type of vehicle, in this case a car, truck or motorcycle and licence_loss means the right to drive on public roads will be revoked.

The determination of the custody of children in Australian family law is considered to be extremely open textured. According to the Family Law Act (1975) the only factor to be taken into account is *the paramount interests of the child*. Following considerable litigation and uncertainty, the Australian Federal Parliament made minimal attempts to define what are the paramount interests of a child. They did this by identifying in the legislation factors such as education, health, a child’s relationship with both parents, and the need to keep siblings together. But there is no clear list of factors. Indeed, it is much easier to describe what is not in a child’s best interests (for example sexual abuse, or violence) than what is in a child’s best interests. It is clearly impossible to write a set of rules determining the paramount interests of the child.

A predicate is bounded if the problem space can be specified in advance, regardless of the final definitional interpretation of the terms in the problem space. A problem space is unbounded if one cannot specify in advance which terms lie within the problem space. Stranieri et al. [28] concluded that legal domains could be divided into four quadrants depending upon their degree of boundedness and open texture. Fig. 1 indicates these quadrants.

Gorry and Scott-Morton [14] define unstructured decisions as those in which the decision maker must provide judgment, evaluation and insights into the problem definition. Structured decisions are repetitive, routine and involve a definite procedure for handling decision making in such domains. Zeleznikow and Hunter [40] argue that one can use rule-based systems to model structured domains. However, other techniques must be used in unstructured domains—especially one in which the decision maker has been allowed to exercise his/her discretion. We believe that narrow-bounded domains can be modeled using rule-based systems, whereas it is not feasible to model wide unbounded domains (and thus we have not attempted to build decision support systems that advise upon child custody).

The dimension open-textured–well-defined refers to our belief as to the extent to which a task is open textured. Although every possible extension for an open-textured concept cannot be predicted, we believe that it is possible to estimate the extent to which the known extensions represent all possibilities. Practitioners seem to estimate the degree of open texture of a statute in order to offer a prediction.

We shall illustrate these principles by considering some examples from the domain of Australian Family Law. For example, the concept of *liability to pay child support* under the Child Support Act (1988) is far less subject to new uses than the concept of *paramount interests of the child*, which is the sole criterion in determining the custody of and access to children. The Child Support Act (1988) specifies the financial liability of a non-custodial parent for his children. The formula is a function of both parents' incomes and the number of children and other dependents both parents have.

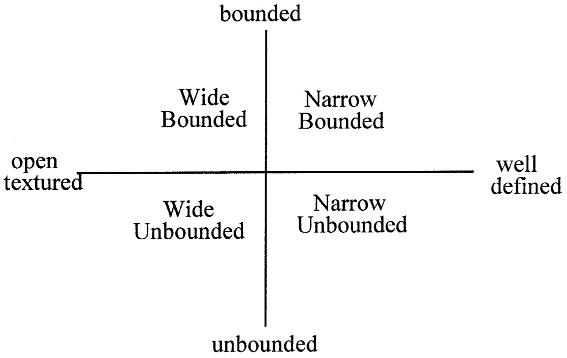

Fig. 1. Quadrants for classifying open texturedness and boundedness of legal domains.

The bounded–unbounded dimension refers to the extent to which an expert believes that all terms relevant for the performance of a task are explicitly known. Because we are confident about what factors are involved in both common pool determination and the percentage split determination (see later), we claim both tasks are bounded. The task of predicting custody arrangements is quite unbounded since we do not believe all, or even most, factors relevant for this determination are known. Each judge has her/his own set of family values, which cannot be automated.

In determining the distribution of property under the Family Law Act (1975) a judge performs the following functions:

1. She determines the assets of the marriage the court is empowered to distribute. This task is known as common pool determination.
2. She determines what percentage of the common pool each party is empowered to receive.
3. She determines a final property order in line with the decisions made in 1 and 2.

The task of creating property orders (following the common pool and percentage split determination) is also unbounded (although it is not open textured). Few features relevant for this task are known, though judges generally avoid forcing a sale of any asset and they also attempt to minimise the disruption to the everyday life of children. There are no other obvious relevant factors or heuristics. The statute provides no guidance and there have been very few litigated cases that specifically relate to the court order created.

Tasks that fall in the narrow-bounded quadrant (top right in Fig. 1) are well suited to implementation with rule-based reasoning or within a logic programming paradigm. Logic limits its inferences to deduction and cannot represent uncertainty. But these limitations are not restrictive for narrow bounded tasks. A representation of uncertainty is not required here because the terms relevant for a solution are known, as is the manner in which these terms combine. The common pool determination was thus implemented as a rule-based reasoner. We may summarise these findings in Table 2.

The Split-Up system implements steps 1 and 2 above, the common pool determination and the prediction of a percentage split. According to domain experts, the common pool determination task (step 1) does not greatly involve the exercise of discretion, in stark contrast to the percentage split task (step 2). Consequently, Split-Up implements the common pool determination by eliciting heuristics as directed graphs from domain experts using a methodology we have called sequenced transition networks. This approach is described in Section 3 of this paper. In that section we will also describe our use of neural networks and rules for the second task above, the determination of a percentage split of marital assets.

In this paper we shall focus upon providing intelligent decision support in domains represented by uncertainty. Rule-based reasoning is thus inadequate. In addition to using fuzzy logic and probabilistic reasoning, it is useful to reason with data. Case-based reasoning proves useful in dealing with such problems. Ashley [1] states that case-based reasoning is particularly useful in (a) interpreting rules, (b) supplementing weak domain models and (c) supporting knowledge acquisition and learning. Recently, data mining and knowledge discovery in databases techniques (KDD) have been developed to learn from data in semi-structured and unstructured domain domains.

To use KDD techniques we require the existence of commonplace cases [45]. Within law, those decisions from appellate courts that form the basis of later decision and provide guidance to lower courts do provide a fundamental lesson, or normative structure for subsequent reasoning. The common name for such cases is landmark cases. Most decisions in any jurisdiction are not landmark cases. They are commonplace, and deal with relatively minor matters such as vehicle accidents, small civil actions and petty crime. These cases are rarely, if ever, reported upon by court reporting services. More importantly, each case does not have the same consequences as the landmark cases. Landmark cases are therefore of a fundamentally different character to commonplace cases. They will individually have a profound effect on the subsequent disposition of all cases in that domain, whereas commonplace cases will only have a cumulative effect, and that effect will only be apparent over time. When learning from cases (as we shall demonstrate when considering property division in Australian Family Law) we need to use commonplace cases.

We now consider how to use knowledge discovery from databases to understand how Australian Family Court judges exercise discretion when distributing property following divorce.

Table 2

Task	Open textured–well defined	Bounded–unbounded	Quadrant
Determining whether an asset is to be placed in the common pool	Well defined; most of the Act comprises definitions of terms used within the Act	Bounded; no discretionary provisions; judges follow leading cases	Narrow bounded
Creating a property order	A few open-textured terms; but no discretionary provisions	Unbounded; no list of factors	Narrow unbounded
Determining custody of a child	Many open-textured terms; prime one is the paramount interests of the child	The decision maker is allowed a great deal of discretion and no bound on the number of factors	Wide unbounded
Percentage split determination	Many open-textured terms	Bounded; definitions cannot be modified	Wide bounded

3. The Split-Up system: reasoning and learning in domains characterised by uncertainty

KDD is an emerging field combining techniques from databases, statistics and artificial intelligence, which is concerned with the theoretical and practical issues of extracting high level information (or knowledge) from a large volume of low-level data. Fayyad et al. [11] define knowledge discovery in databases (KDD) as ‘the non-trivial process of identifying valid, novel, potentially useful understandable patterns in data’. Because most KDD systems use some form of statistical algorithm to discover knowledge, data mining and knowledge discovery systems fail to provide adequate explanation—an essential element of any decision support system. In law and the social sciences an explanation of the system’s reasoning can be as important as the decision reached. For this reason, whilst cases have been regularly used in building legal case-based reasoners, they have rarely been used as a means of automated discovery of legal knowledge. Knowledge discovery techniques in law require some manual analysis of the data and the KDD process can only provide support for legal practitioners if commonplace cases are abundant.

3.1. Sequenced transition networks

The sequenced transition network methodology (STN) enables the automated translation of a directed graph into sets where each set represents a path within the graph [27]. Four set operators defined in the STN approach are applied to the sets in order to implement forward chaining, backward chaining and the generation of explanations. This approach is conceptually equivalent to rule-based reasoning although the role of a knowledge engineer is kept to a minimum. Using this methodology, knowledge acquisition and maintenance benefits result because rules are not required at all. There is no requirement to convert graphs to rules because directed graphs drawn by the expert are automatically converted into sets. Fig. 2 illustrates a directed graph that represents the interaction between a family law expert and a client.

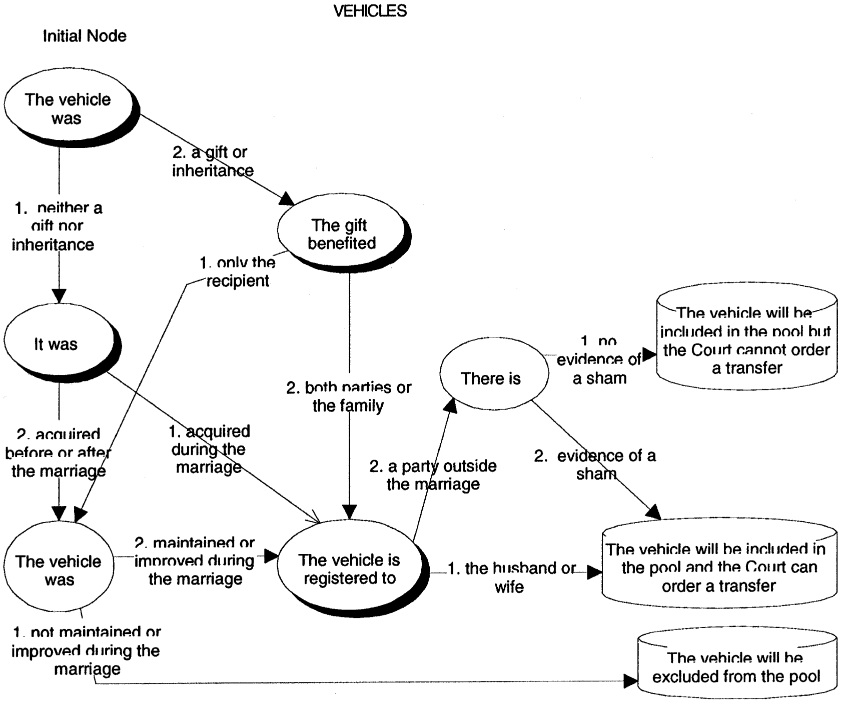
VEHICLES
Initial Node
The vehicle was
2. a gift or inheritance
1. neither a gift nor inheritance
The gift benefited
1. only the recipient
It was
2. both parties or the family
There is
1. no evidence of a sham
The vehicle will be included in the pool but the Court cannot order a transfer
2. acquired before or after the marriage
1. acquired during the marriage
2. a party outside the marriage
2. evidence of a sham
The vehicle was
2. maintained or improved during the marriage
The vehicle is registered to
1. the husband or wife
The vehicle will be included in the pool and the Court can order a transfer
1. not maintained or improved during the marriage
The vehicle will be excluded from the pool

Fig. 2. Directed graph for common pool determination.

The principal domain expert for Split-Up was Renata Alexander, an experienced family law practitioner with a government funded legal agency in the State of Victoria. The graph of Fig. 2 is one of 51 she drew to capture knowledge relevant for determining whether an asset will be considered by the court or not. An STN program labels each node in the graph. The initial node is labelled 0. The node reached by traversing arc 2 from the initial node is labelled node 02. If a node in the graph is reached by more than one path, then the node receives two labels. The program stores each node as an object that is identified by the node label. Text associated with the node and with arcs emanating from the node on the graph is also stored within the object for subsequent use as prompts and explanations. Thus, a directed graph is not translated into rules for inference engine utilisation as is the case in traditional rule-based expert system design. No prompts, questions or text other than what is drawn on the graph are required.

Forward chaining inferencing commences with the presentation to the user of the text associated with the initial node as a user prompt. The user is then provided with a number of response options. The arc number of the user's response is appended to the node number to retrieve the next node. This proceeds until a conclusion is reached.

The STN methodology is an approach equivalent to a rule-based approach but has some acquisition and maintenance benefits. The majority of rule-based systems use proprietary expert systems shells, which are often expensive and available only on limited platforms.

Although research in intelligent systems for law over recent years has focused on sophisticated reasoning methods such as argumentation and case-based reasoning, only the earlier and less ambitious rule-based reasoning systems have been adopted for commercial use. However, the majority of rule-based systems in law use proprietary expert systems shells, which are often expensive and available only on limited platforms. Few rule-based systems have been developed for use directly with World Wide Web technology. Furthermore, few rule-based reasoning systems are simple enough to be used by domain experts; a feature that is critical given the high cost of knowledge engineers and well reported knowledge acquisition problems. The methodology has been used for the acquisition of knowledge relating to the distribution of property following a divorce in Australian family law. We are currently using the STN methodology to build rule-based systems in the following domains: (i) eligibility for legal aid; (ii) modeling the Child Support Act; and (iii) copyright issues related to computer software.

3.2. Percentage split determination

The Family Law Act (1975) directs a decision maker to take into account the past contributions of each party to a failed marriage in addition to their resources for coping with life into the future. Rather than offering one definition for *contributions* and one for *needs*, the statute presents a ‘shopping list’ of factors to be taken into account in arriving at a property order. For example, the age, state of health and financial resources are explicitly mentioned in the statute as relevant factors, yet their relative levels of importance are unspecified.

Although the statute presents a flat list of relevant factors without specifying how these factors relate to each other, we realised that the factors could be placed in a hierarchy. The development of the hierarchy required specific knowledge supplied by domain experts. A hierarchy of 94 factors presented in Fig. 3 was elicited. Fig. 3 demonstrates that the factors relevant for a percentage split determination (extreme right of figure) are past contributions of a husband relative to those of the wife, the husband's future needs relative to those of the wife, and the wealth of the marriage. The factors relevant for a determination of past contributions are the relative direct and indirect contributions of both parties, the length of the marriage and the relative contributions of both parties to the homemaking role. No attempt is made in Fig. 3 to represent the way in which relevant factors combine to infer factors higher in the hierarchy. The hierarchy of Fig. 3 provides a structure that was used to decompose the task of predicting an outcome into 35 sub-tasks. Outputs of sub-tasks further down the hierarchy are used as inputs into sub-tasks higher in the hierarchy. Solid arcs in Fig. 3 represent inferences performed with the use of rule sets whereas dashed arcs depict inferences performed using neural networks.

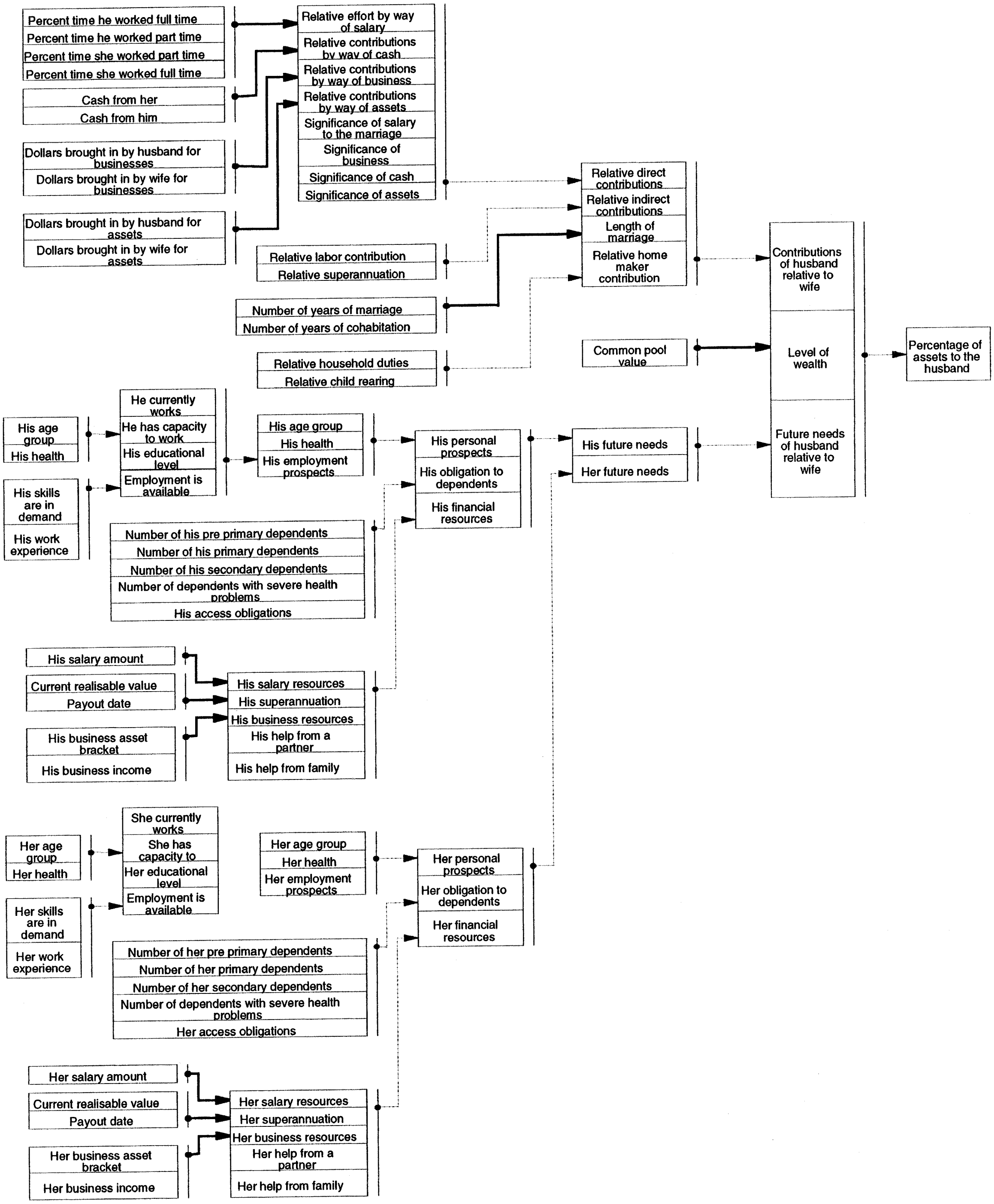
Percent time he worked full time
Percent time he worked part time
Percent time she worked part time
Percent time she worked full time
Cash from her
Cash from him
Dollars brought in by husband for businesses
Dollars brought in by wife for businesses
Dollars brought in by husband for assets
Dollars brought in by wife for assets
Relative effort by way of salary
Relative contributions by way of cash
Relative contributions by way of business
Relative contributions by way of assets
Significance of salary to the marriage
Significance of business
Significance of cash
Significance of assets
Relative labor contribution
Relative superannuation
Number of years of marriage
Number of years of cohabitation
Relative household duties
Relative child rearing
Relative direct contributions
Relative indirect contributions
Length of marriage
Relative home maker contribution
Common pool value
Contributions of husband relative to wife
Level of wealth
Future needs of husband relative to wife
Percentage of assets to the husband
His age group
His health
His skills are in demand
His work experience
He currently works
He has capacity to work
His educational level
Employment is available
His age group
His health
His employment prospects
His personal prospects
His obligation to dependents
His financial resources
His future needs
Her future needs
Number of his pre primary dependents
Number of his primary dependents
Number of his secondary dependents
Number of dependents with severe health problems
His access obligations
His salary amount
Current realisable value
Payout date
His business asset bracket
His business income
His salary resources
His superannuation
His business resources
His help from a partner
His help from family
Her age group
Her health
Her skills are in demand
Her work experience
She currently works
She has capacity to
Her educational level
Employment is available
Her age group
Her health
Her employment prospects
Her personal prospects
Her obligation to dependents
Her financial resources
Number of her pre primary dependents
Number of her primary dependents
Number of her secondary dependents
Number of dependents with severe health problems
Her access obligations
Her salary amount
Current realisable value
Payout date
Her business asset bracket
Her business income
Her salary resources
Her superannuation
Her business resources
Her help from a partner
Her help from family

Fig. 3. Hierarchy of relevant factors for percentage split determination.

Fig. 4 illustrates the framework for inferring a percentage split outcome with the use of a neural network. This figure expands the factors on the right of Fig. 3. The inputs to the neural network (depicted

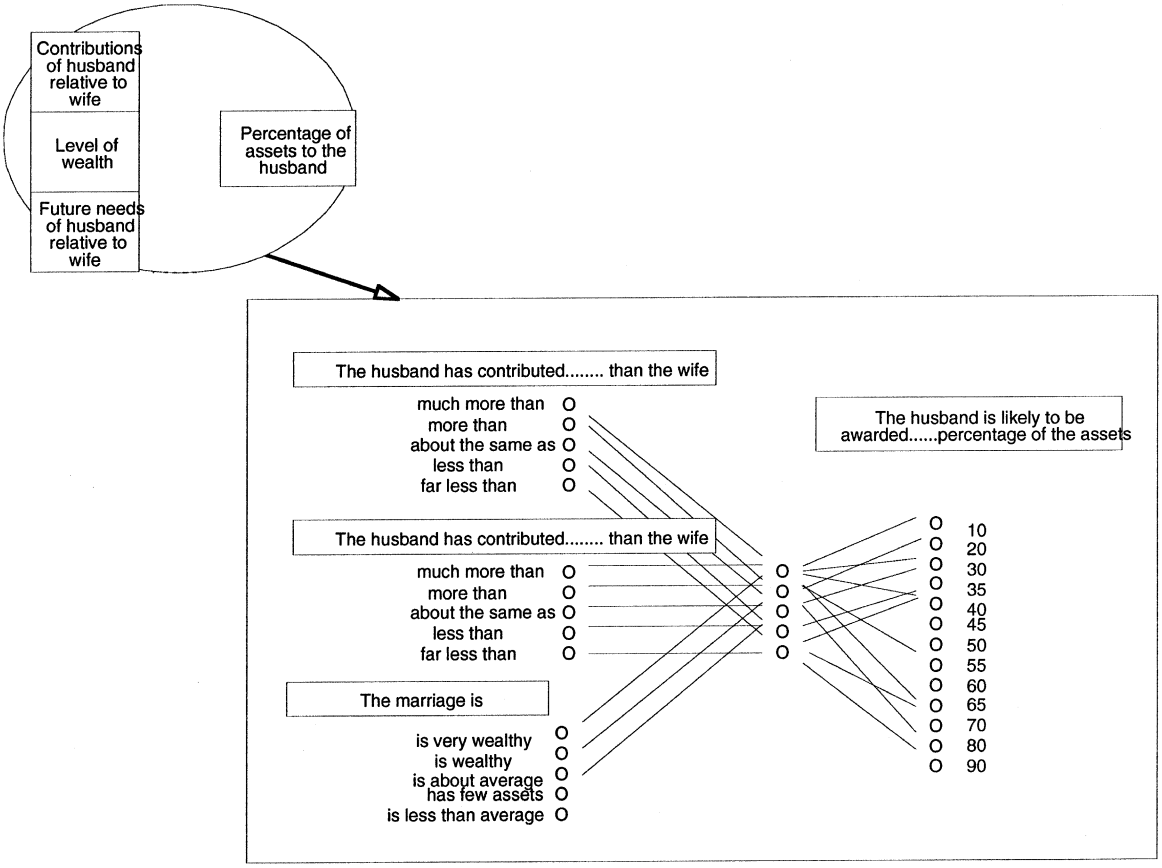
Contributions of husband relative to wife
Level of wealth
Future needs of husband relative to wife
Percentage of assets to the husband
The husband has contributed........ than the wife
much more than
more than
about the same as
less than
far less than
The husband has contributed........ than the wife
much more than
more than
about the same as
less than
far less than
The marriage is
is very wealthy
is wealthy
is about average
has few assets
is less than average
The husband is likely to be awarded......percentage of the assets
10
20
30
35
40
45
50
55
60
65
70
80
90

Fig. 4. Inferring a percentage split outcome with a neural network.

on the left edge of Fig. 4) are values on each of the three relevant factors, contributions, future needs and wealth. The neural network's output (on the right) is the value of the percentage split predicted. The inferencing of 20 sub-tasks was performed each with its own neural network, whilst for the remaining 15 sub-tasks, small rule sets were used.

As mentioned earlier, the principal obstacle to the use of neural networks in the legal domain is that explanations for inferences cannot be directly generated from the inferencing process. We have overcome this problem by embedding the neural network within a knowledge representation framework based on the structure of arguments proposed by Toulmin [31].

Neural networks operate by gleaning from the training set the weights of the various factors that lead to a certain decision. The manner in which these weights are learned is primarily statistical. In contrast to rule-based reasoning, connectionism is well suited to modeling discretionary reasoning [42]. Domain knowledge of legal rules and principles is not modeled directly; nor are a select number of landmark cases retrieved and adapted as occurs when using case-based reasoning. Instead, a connectionist learning algorithm is exposed to data from a large number of cases previously decided so that the way judges have actually exercised discretion in weighing relevant factors can be assimilated into the program.

Neural networks have rarely been used in the legal domain because explanations are difficult to generate and assembling training sets of sufficient size and coverage is similarly difficult. Our approach has been that connectionism can be useful in law if a series of smaller, interconnected networks are used instead of one larger network and if explanations are generated independently of the process used to infer a conclusion. To provide explanation independently of the conclusion inferred, we used Toulmin Argument Structures.

Factors that were relevant in determining a percentage split in the Split-Up system were elicited from experts and from statutes to form a hierarchy of relevant factors. The argument-based framework used in Split-Up is not limited to rules and neural networks but can easily accommodate other forms of inferencing including fuzzy logic, inferential statistics and non-monotonic logic.

3.3. Knowledge discovery in the Split-Up system

Fayyad et al. [11] highlight that effective knowledge discovery involves a number of steps prior to the application of neural networks. The first phase of any KDD process involves the selection of a sample of data from a store of real world data. In the next phase the data must be pre-processed to remove excessive noise and mistakes. Data is further required to be transformed so that spurious attributes do not clutter the learning algorithm. Neural networks and rule induction are techniques that can be applied in the next phase, known as data mining. Each phase of knowledge discovery in Split-Up has required assumptions that we believe are applicable to knowledge discovery in legal fields other than family law.

3.3.1. Phase 1 gathering raw data

In order to discover how judges weigh different factors, we use, as source material, written judgments handed down by judicial decision makers in commonplace cases. 103 cases involved property alone. Three raters extracted data from these cases by reading the text of the judgment and recording values of 94 template variables. Inter-rater agreement tests were performed informally. Any variable that seemed ambiguous or unclear was highlighted so that a consensus could be reached between the raters.

Data for the Split-Up project was gathered from cases decided between 1992 and 1994. Each of the cases examined had been decided by one of eight different judges. Judgments from these eight judges were examined in preference to limiting ourselves to those from only one judge in order to encourage the network to mimic a composite of all judges.

3.3.2. Phase 2 pre-processing raw data

Data from the domain of property division within Australian family law differs from many other domains in that we expect contradictions. For the purpose of our work, we define the term, thus: *Two cases are contradictory if their inputs are identical yet their outputs differ*. Contradictions are expected because the weighting of factors can vary between judges and within the same judge over time. Thus, two cases could be recorded with the same input set values but different output values.

Contradictory cases are necessarily present in discretionary domains because judges cannot be expected to weight factors in the same way on every case throughout their career, and they cannot be expected to be perfectly consistent with the weightings other judges use. Although contradictory examples are expected in this discretionary domain they should not simply be ignored when training neural networks. A simple example may illustrate this. Consider two cases, A and B, that have identical inputs yet case A resulted in a 70% determination and case B (made perhaps erroneously by a different judge) resulted in 40%. A network trained only with these cases, and presented with identical inputs, will output a value intermediate between the outputs; in this case 55%. The intermediate result of 55% is unacceptable to us. The following are a number of ways to deal with extreme contradictions.

(1) Ignore the extreme contradictions. If sufficient data is collected, then the majority of typical outputs will outweigh the effects of a handful of extreme cases. This strategy is acceptable though relies on the existence of quite large data sets for network training. Given the limited sample size, we opted against this strategy.

(2) Modify one or more contradictory examples to remove the anomaly. This is tantamount to inventing data and was not done.

(3) Remove extreme contradictions from the training set. This is the strategy we have adopted in this study but we do note that this is not without ramifications.

Although the removal of extreme cases from training sets is necessarily a subjective exercise, we can implement a degree of consistency in our method by designing a metric that discerns the extent to which two outcomes are contradictory. The metric we have used in Split-Up relies on the representation of all inputs and outputs as binary digits. For example, the percentage split neural network output is not one output that can take any value between 0 and 100 but is, instead, 13 separate outputs each of which can take the value 0 or 1. The same network has 15 binary inputs that represent one of five possible values on three variables.

Two binary outcomes can be compared by noting the position of the set bit in each outcome. Thus, an outcome of 1 0 0 0 0 differs from one represented by 0 0 0 0 1 by four place units. The second set bit is four places away from the set bit in the former outcome. We call this a four-place contradiction.

In all networks in Split-Up we have removed all examples that have identical inputs but differ from each other by outputs three or more places apart. This criteria is necessarily subjective. Allowing extreme contradictions to remain in the training set is unwise, yet determining which contradictions ought to be labelled extreme is not straight forward.

3.3.3. Phase 3 transformation data

The third phase of the knowledge discovery process involves transforming the processed data set to a form likely to be most fruitful. This phase in Split-Up involves the decomposition of the task into 35 sub-tasks according to the hierarchy of arguments depicted in Fig. 3. Each sub-task could thus be treated as a separate (and smaller) data mining exercise. This decomposition also enabled each set of examples to be free of null values.

3.3.4. Phase 4 data mining using neural networks

Data mining was performed in Split-Up with the use of neural networks. There are many types of network that could have been used, though we restrict ourselves to feed forward networks trained with backpropagation of errors. Feed forward neural networks trained with backpropogation learning are said to generalise well if the output of the network is correct (or nearly correct) for examples not seen during training.

To our knowledge, no application of neural networks in law, using real or hypothetical data, have employed techniques to ensure that a trained network reflects patterns in the actual population of cases and is not merely a reflection of the sample data gathered. Weiss and Kulikowski [33] provide a comprehensive overview of statistical techniques that help to ensure a trained network represents characteristics of the entire population and is not an aberration linked solely to the particular set of cases selected for training and testing. They note that few classifiers remain generally accepted unless some effort has been made to evaluate the performance of the classifier on an entire population and not just on a sample of data.

3.4. Argumentation in the Split-Up system

Toulmin [31] concluded that all arguments consist of four invariants: claim, data, warrant and backing. A detailed description of how Toulmin argument structures are used in Split-Up can be found in Ref. [26]. An excellent review of how Toulmin's theory has been used to build intelligent decision support systems can be found in Ref. [25].

Toulmin states that the assertion of an argument stands as the claim of the argument. Knowing the data and the claim does not necessarily convince one that the claim follows from the data. A mechanism is required to justify the claim given the data. This justification is known as the warrant. The backing of an argument supports the validity of the warrant. In the legal domain it is typically a reference to a statute or a precedent.

Fig. 5 represents the complete argument structure for the percentage split argument. Note that only the claim and data can be represented in the hierarchy of Fig. 3.

The reason that the data item ‘The husband has contributed more to the marriage’ is relevant in the percentage split argument within Split-Up is that Section 79 (4) of the Family Law Act specifically obliges a decision maker to take past contributions into account.

Explicitly representing the inference method enables the use of a variety of artificial intelligence inferencing procedures. For example, rules are used to infer assertions in Split-Up for some arguments

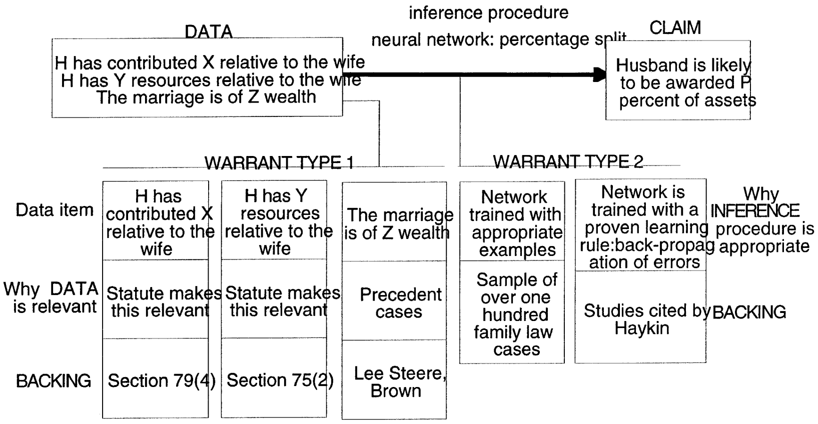
inference procedure
neural network: percentage split
DATA
H has contributed X relative to the wife
H has Y resources relative to the wife
The marriage is of Z wealth
CLAIM
Husband is likely to be awarded P percent of assets
WARRANT TYPE 1
WARRANT TYPE 2
Data item
H has contributed X relative to the wife
H has Y resources relative to the wife
The marriage is of Z wealth
Network trained with appropriate examples
Network is trained with a proven learning rule:back-propagation of errors
Why INFERENCE procedure is appropriate
Why DATA is relevant
Statute makes this relevant
Statute makes this relevant
Precedent cases
Sample of over one hundred family law cases
Studies cited by Haykin
BACKING
BACKING
Section 79(4)
Section 75(2)
Lee Steere, Brown

Fig. 5. Argument structure used in Split-Up.

whilst neural networks are used for others. Explicitly representing the inference method used in an argument enables us to clearly specify which type of inference has been used for each argument. Argument claims can follow from data by deduction, induction or analogy. The original Toulmin formulation does not permit the specification of the type of inference in use within a particular argument. Knowing the type of inference is important in our efforts to accept or rebut an argument.

A reason that explains why an inference procedure is appropriate is a form of warrant. This contributes to an explanation of why a claim follows from data. As Fig. 5 illustrates, the neural network used in the percentage split argument is suitable because it has been trained with data from 101 actual cases. Additionally, it is appropriate that it was trained with a proven learning rule. Conversely, a reason that the application of a rule is appropriate in other arguments is that the inference is an instance of modus ponens, an inference rule that is demonstrably sound.

Thirty-five arguments were identified in consultation with domain experts for the determination of an appropriate percentage split of the assets of a marriage. In asking our experts to develop each argument structure, we are not eliciting heuristics because a claim is inferred from data within an argument structure by a neural network trained with cases and not by domain expert heuristics. However, ascertaining which elements are relevant for each argument was determined by domain experts.

The Toulmin argument structure enabled us to decompose the task of determining a percentage split outcome into 64 sub-tasks where each sub-task represents an argument. Many of these arguments produced claims that were in turn used as data for other arguments. All arguments contribute to a culminating argument—the percentage split illustrated on the right of Fig. 3.

The claim of each argument is inferred from data values from the same argument. The inference for an argument is performed by feeding data values forward through a neural network associated with that argument. Most neural networks are small because the entire task has been decomposed into smaller sub-tasks.

The hierarchy of arguments is critical in decomposing the task into smaller sub-tasks in the data transformation phase so that data mining techniques can be applied more effectively to each sub-task. In Ref. [20] we used the 94 attributes in the Split-Up data in a flat structure without segregating the attributes into independent arguments and found that the best data mining methods we used were only able to predict 35% of case outcomes.

The generation of an explanation commences once a claim has been inferred. The user may question this claim. The data items that were involved in inferring the claim are then presented as an initial explanation. If the user cannot accept the data item value as valid, the argument that produced those items is found and an explanation is generated for it. If the validity of the data items is not in question but the rationale is questioned, the warrant of the argument is produced. This is augmented with the backing if the user is still dissatisfied.

An explanation generated in this way is independent of the inferencing method used to produce the claim. Thus, an explanation can be generated whether a rule set, or a neural network or any other inferencing method had been used to produce the claim. The explanations are implemented in Split-Up as hypertext links to Toulmin argument components. The percentage split module of Split-Up has been implemented using the object oriented knowledge-based system development tool, KnowledgePro. The hypertext facilities built into KnowledgePro allow the warrant-and backing-based explanations to draw on statutes and past cases. Those arguments that are rule-based make use of KnowledgePro's forward and backward chaining inferencing facilities Neural network-based arguments call on Split-Up's facilities to determine the claim for an argument.

3.5. Implementing the Split-Up system

Split-Up has been implemented using KnowledgePro as an argument-based reasoning shell (KnowledgePro is an object-oriented high level language with a built-in inference engine and hypermedia development tools released by Knowledge Garden and runs on a PC-Windows environment). Family law knowledge has been entered into the shell so that the argument-based framework can be evaluated, though studies are under way to demonstrate that the shell can also be useful applied within non-legal domains. The basic unit of knowledge in Split-Up is the sentence. All data, claim, warrant and backing items are sentences. All sentences are stored in a sentence base and are retrieved to produce user prompts, claims and explanations. Arguments are frames with slots that reference sentences in a sentence base.

Software used for neural network training was neuDL (Neural Network Description Language), a description language for the design, training and operation of neural networks developed at the University of Alabama. Using this language rather than specialised network software enabled us to implement our own performance metrics described above. All neural networks were trained on mainframe computers running Unix for speed and efficiency.

Invoking a number of neural networks for a single consultation presents an efficiency problem. Fig. 6 represents the system design typically used when a number of different neural networks are called upon to perform an inference. This figure illustrates that once trained, each network's topology, weights, biases and activation function are stored in a repository. A run time invocation of a network requires, in essence, that the network be rebuilt with information retrieved from the repository before the input data can be fed through to produce an output. This can seriously degrade performance in a system such as Split-Up that consecutively invokes over 20 networks. Rather than include the neural network functions in Split Up, we have captured the results of inferencing in a data structure.

Each possible input is presented to the trained network prior to a consultation in a pre-specified order. The position of a given input in this ordering

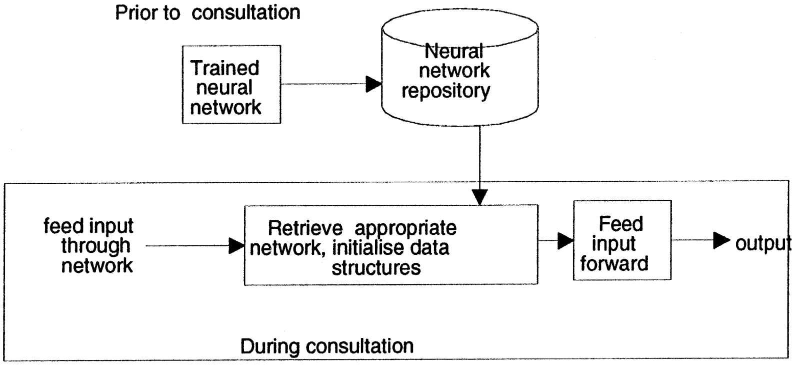
Prior to consultation
Trained neural network
Neural network repository
feed input through network
Retrieve appropriate network, initialise data structures
Feed input forward
output
During consultation

Fig. 6. Schema for typical neural systems.

is determinable solely from the input's value. The outputs are stored in the order they emerge from the network, so that the outputs are also ordered. To find the output that corresponds to given input, the position of the input is determined and the output at that position in the output list is retrieved.

This approach is illustrated in Fig. 7. There is no need to store information about each network other than the sequence of outputs. Rebuilding a network is unnecessary, nor is there a need to actually feed values through a network during consultation. Instead the output value that corresponds to the input is retrieved from the ordered list of outputs. These factors combine to greatly enhance the efficiency of the system.

3.6. Explanation in the Split-Up system

Split-Up explains its reasoning for inferring an argument's assertion by presenting the data, warrant and backing components of the argument to the user on request. For example, if the user invokes an explanation for the assertion 'Overall, the husband is likely to be awarded 40% of the assets' he/she is presented with the data items from the argument structure 'The husband has contributed to the same extent as the wife, The husband has greater resources for the future as the wife, The marriage is of average wealth'. If any one of these data items is questioned by the user, the argument that produced the data item as an assertion is retrieved and an explanation generated from it. If, on the other hand, the user is satisfied with the data items but wants further explanation, the reasons for the relevance of each data item and the reason for the appropriateness of the inference method are retrieved from the argument structure: 'Contributions must be taken into account according to the statute: 79 (4). Resources for the future must be taken into account according to the statute: 79 (4)e 75 (2). Inference has been produced using a neural network trained with appropriate examples: over 100 real Family Court cases. The neural network was trained using backpropagation of errors: a proven learning algorithm'.

Eberhart [10] claims that the purpose of an explanation facility within an expert system is to encourage the user to trust the system as opposed to the purpose of rule-based system explanation facilities which was to aid knowledge engineers to debug large rule sets. The early 'trace' type of explanation facility reflected the inferencing process perfectly though typically did not engender a user's trust. In a similar vein, Bench-Capon et al. [5] noted that explanations were more than proof procedures and reported favourable user responses when they used Toulmin argument structures to provide explanations for their logic programs. Wick and Thompson [35] also note that explanation involves more than the reproduction of inferencing steps. They have developed an explication facility that is invoked after the inferencing has concluded. It takes, as input, the inferencing steps used to reach a conclusion, in addition to domain knowledge at a different level of specificity to that used to infer conclusions.

Generating explanations for neural network inferences is difficult because the inferencing steps are not explicit. Nevertheless, two broad approaches that have emerged aim to generate explanations in this paradigm. One approach involves selecting a sample of examples that most closely matches the input.

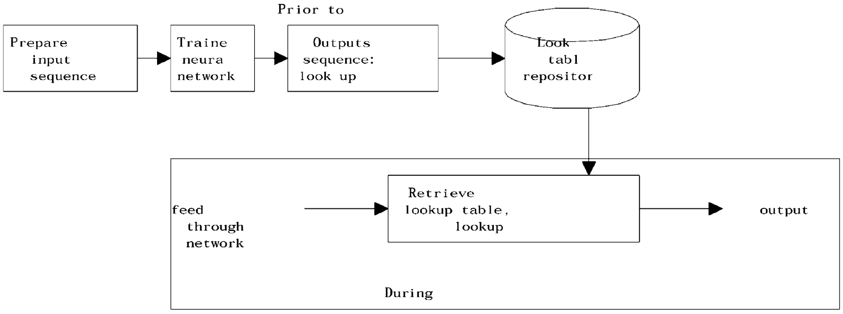

Fig. 7. Schematic diagram of lookup table repository method.

These examples are presented to the user as similar cases. This approach certainly has its uses but can be limited in law. A sample of similar examples does not make explicit a statute or precedent case that underlies many inferences. A different approach in neural network explanation, exemplified by Diederich [9], involves representing the internal sub-symbolic processes within a neural net in a symbolic manner so that inference steps can be elucidated. While useful, these approaches are limited in law because an explanation that will engender trust must provide information over and above that involved in inferencing steps.

Explanations in Split-Up are pragmatically grounded because they are supplied by domain experts. There is no suggestion that the reason for the relevance of a data item specified by domain experts is the only one possible, nor is it necessarily the ideal reason. It is, however, a reason that makes pragmatic sense to the expert. As such, it is more likely to engender the user's trust than if a reason that replicated reasoning steps was used.

4. Current research involved in developing Split-Up into a commercial tool

In building negotiation support tools we have assumed that all actors behave *rationally*. Principled negotiation [13] promotes deciding issues on their merits rather than through a haggling process focused on what each side says it will and will not do. It promotes a focus on interests of the party rather than allowing negotiation to deteriorate into a contest of 'who will back down first'. Fundamental to the concept of principled negotiation is the notion of *Know your best alternative to a negotiated agreement (BATNA)*. The reason you negotiate with someone is to produce better results than would otherwise occur. If you are unaware of what results you could obtain if the negotiations are unsuccessful, you run the risk of (1) entering into an agreement that you would be better off rejecting, or (2) rejecting an agreement you would be better off entering into. Sycara [29,30] notes that in developing real world negotiation support systems, one must assume bounded rationality and the presence of incomplete information. Many developers of negotiation support systems use game theory. Similar to Sycara, we prefer to use artificial intelligence techniques.

Given our belief that it is imperative to focus upon specific negotiation domains; at La Trobe University we have commenced our research on building intelligent negotiation support systems by considering disputes in Australian Family Law. In Ref. [2] we developed Family_Negotiator, a hybrid rule-based/case-based intelligent DSS that supports negotiation in Australian Family Law. In Ref. [4] we extended the Family_Negotiator system with game theory techniques and fuzzy cognitive maps.

An important way mediators encourage disputants to resolve their conflicts is through the use of compromise and trade-offs. It is thus imperative to focus on the relationships between each of the issues in dispute in order to establish the best method of obtaining a satisfactory settlement [3]. Wellman [34] states that most of the decisions we make are trade-off situations. Once the issues have been identified, other decision-making mechanisms must be employed to resolve the dispute. An assumption we are making in modeling negotiation is that dependencies between issues exist to degrees specified by the disputants. Points given by users are analysed to form degrees to which these dependencies exist (otherwise referred to as trade-off degrees). The algorithm discussed later in this paper introduces the reader to the concept of hierarchical decomposition of issues. It is a representation that lists all issues and their sub-issues with hierarchical links intact. Sub-issues (identified in the decomposition hierarchy) and trade-offs degrees are used to form cognitive maps. A particular form of cognitive mapping, to be known as a bi-directional fuzzy cognitive map (BFCM), was introduced and demonstrated by Bellucci and Zeleznikow [4].

In the Split-Up system we have used knowledge discovery techniques to determine how Australian Family Court judges use their discretion to distribute marital property. While Split-Up can be used to determine one's BATNA for a negotiation, it does not model the negotiation process itself. Split-Up first shows both litigants what they would be expected to be awarded by a court if their relative claims were accepted. It gives them relevant advice as to what would happen if some, or all of their claims were rejected. They are able to have dialogues with the Split-Up system about hypothetical situations that would support their negotiation. Both litigants then have clear ideas about the strengths and weakness of their claims.

4.1. Commercialising the Split-Up system

Despite the Split-Up system receiving much media publicity, we have received only minimal assistance in commercialising the system. Our research has shown that it is feasible to build legal decision support systems that learn from commonplace cases. However, the current version of Split-Up is a research prototype. To build a decision support system that is widely used by legal practitioners, we must (a) develop a much better user interface, (b) ensure the system is more robust, and (c) add hundreds more recent commonplace cases to the system. Recently, the Australian Research Council, through its SPIRT (Strategic Partnership in Research and Training) grants has given the Split-Up project a large 3-year grant. Together with Victorian Legal Aid and Phillips and Wilkins Solicitors, the project will receive US$500,000 to conduct research and eventually build a commercial system using fuzzy logic. At the conclusion of the project, when the case base is increased 10-fold, evaluation is rigorously undertaken and alternative KDD techniques explored, we expect to have a robust commercial system.

An essential component in the development of legal decision support systems in discretionary domains is the evaluation of the effectiveness of such systems. It is thus imperative to design, develop and implement a strategy for evaluating the effectiveness of programs that perform inferences on the basis of knowledge discovered from KDD techniques including (a) an empirical study comparing user satisfaction between four groups of users that have different information needs (registrars, judges, mediators and lawyers) and (b) the discovery/development of an instrument for the evaluation of the efficacy of explanations offered by computer programs in the legal domain.

We are using techniques of Reich [19] to evaluate legal expert systems [23].

4.2. Feature selection for the Split-Up system

Currently, following the advice of domain experts, the Split-Up system uses 94 different attributes. The Split-Up architecture provides no mechanism for determining whether the factors are relevant in empirical terms. It is possible that many of the factors declared relevant by our experts do not, in practice, contribute to a prediction. Thus, a family law prediction could possibly be made with only a subset of the factors regarded as relevant by experts.

We have applied feature selection techniques using genetic search to the data used to determine percentage split in the Split-Up system [20]. We have used genetic algorithms to determine which attributes are essential to model when distributing marital property. Our research shows a more accurate prediction can be made when using 16 of the 94 variables. An interpretation of this result is that the other 78 attributes are rarely used by Family Court judges when distributing property.

4.3. Comparing neural networks and other statistical algorithms

As mentioned above, the argument-based framework used in Split-Up is not limited to rules and neural networks but can easily accommodate other forms of inferencing including fuzzy logic, inferential statistics and non-monotonic logic. Our recent research has involved investigating the suitability of using other statistical algorithms as part of the Toulmin Argument Structure for the Split-Up system. We chose one argument, namely the percentage split to the husband, and modeled it using both neural networks and regression analysis [21]. Whilst both algorithms performed similarly for average cases (cases where the husband was awarded between 35% and 65% of the common pool), neural networks proved superior for extreme case. The bias towards the mean displayed by the regression formula in the percentage split argument suggests that a neural network is more appropriate. However, we claim that the argument-based representation adopted here facilitates the use of alternate inferencing methods.

4.4. Dealing with contradictions

In discretionary domains, it is possible that outcomes may still differ even if there are no classification anomalies and the same principles have been used by all judges. Outcomes may be different because judges apply different weights to each relevant factor. Neither judge is wrong at law because the statute clearly affords the decision maker precisely this sort of discretion. The presence of discretion indicates that two or more legitimate judgments may have identical findings of fact yet different outcomes. Dealing with contradictions is thus imperative. Stranieri and Zeleznikow [22] introduce three different forms of stare decisis—traditional, personal and local—so that they can cope with seeming inconsistencies when building Split-Up (like cases should be decided in a like manner).

4.5. The use of statistical techniques in building intelligent soft computing decision support systems

The important distinction between early expert systems and the development of intelligent soft computing decision support systems is the latter's use of statistical techniques.

In the Split-Up domain, expertise is used to identify the relevant factors to distribute property following divorce. Statistical techniques (in the form of neural networks) are then use to learn the relative importance of each of these features. Feature selection algorithms have been used to determine which factors are indeed important. The drawback of all statistical techniques: namely inadequate explanations, has been overcome through the use of Toulmin's theory of argumentation.

In IFDSSEA, domain expertise in the holistic scoring model was used to identify the relevant features needed to properly grade a student essay. Neural networks and rule induction algorithms along with human input were used to identify the relative importance of the features. The results were membership functions for use in a fuzzy logic framework.

5. Intelligent fuzzy decision support systems that provide advice about assessment

Along with evaluation and interpretation, assessment is a fundamental decision making task. Examples of assessment tasks involve evaluating student performance and the viability of various investment options. Often decisions need to be made with insufficient numerical data, or imprecise or vague information [16].

Grading essays is labour intensive, repetitive, and fraught with imprecision. Typically a teacher must learn a scoring standard or 'rubric' that he or she will consistently apply to all student writing samples. Applying the rubric consistently generally takes a considerable amount of time. In addition, the scoring rubrics for writing assessment usually employ the use of linguistic categories and approximate reasoning. This makes it much more difficult to ensure uniform application of the scoring rubrics. Expert decision support help in making the grade decision could lead to quicker evaluation of writing samples and more valid individual and group assessment because the application of the scoring rubrics would be much more uniform.

A popular scoring system used by many school districts (especially in New York State) is the holistic scoring method. This involves evaluating the writing sample for the existence of certain essentials —clarity, mechanics, organisation—and then combining these factors into a grade. When evaluating the essay on these criteria, the grader often uses linguistic criteria such as mostly *clear* or *confused*. Fuzzy logic has been used to reason with the uncertainty and imprecision associated with linguistic variables.

Unfortunately, there is no guarantee that human graders will apply the same rules (or criteria) consistently, nor will there be a high level of consistency between graders. This makes the task of comparing assessments of writing very difficult.

IFDSSEA is an intelligent fuzzy decision support system that assists New York State teachers to assess students' essays. IFDSSEA's primary purpose is to help teachers efficiently and effectively evaluate student essays. It also increases the teacher's application of scoring rules. In addition, IFDSSEA provides less experienced teachers with a tool for developing their essay assessment skills.

The model base of IFDSS consists of two parts: (a) the holistic model base contains a fuzzy reasoning module that enables the user to build assessment decision models using fuzzy logic and holistic scoring principles and (b) the statistical model base contains statistical models for use in classification, pattern recognition and project management. These models are primarily used to classify and manage the writing assessment results.

The inference engine is used to process rules for the application. The database system provides user support for storing, organising and retrieving student assessment. The IFDSS generator acts as a buffer between the user and the other IFDSS components. Front-end support in the form of intelligent assistance to the user when interacting with the assessment knowledge base is given by the IFDBA interface (which allows the user to obtain an explanation of the model results including any reasoning employed). An intelligent user interface assists with all aspects of the natural language communication between the grader and the IFDSS.

Specifically, the development of the expert fuzzy classification system for scoring student writing samples proceeded in the following manner:

1. A group of expert teacher graders was selected and asked to develop ranges of scores corresponding to labels for each of the linguistic feature categories used in the scoring rules.
2. These 'membership' data were used to develop membership function sets for each feature and classification variable.
3. A fuzzy rule base was developed using the New York City Grade 4 PAL test scoring rules. This along with the membership functions, comprised the knowledge-base of the expert fuzzy classification scoring system.

The expert fuzzy classification scoring system was validated by classifying test data sets. This work resulted in the development of 21 membership function sets representing the feature and classification variables. The rulebase consists of over 200 rules. The teacher decides on the ratings to be given for each of the input feature variables, e.g., understanding, recognition of important characters, etc. These ratings are automatically 'fuzzified' and the appropriate rules from the rulebase are fired. The results are 'defuzzified', resulting in numeric scores. The output from the classification component can be visualized and further explained by providing the underlying rules used to make the classification.

A commercially available software package called O'Inca Design Framework was used for developing the membership functions and rules. This fuzzy logic and expert systems shell software package has additional facilities for simulation, on-line modification of rules and membership functions, and displaying output classifications and inference paths.

6. Evaluating intelligent soft computing decision support systems

An important feature in assessing the value of intelligent decision support systems is to assess their outcomes. Until recently, very little emphasis has been placed on evaluating intelligent systems. Recently, artificial intelligence has become an empirical science. Most of the research being conducted under the SPIRT grant discussed in Section 4 is dedicated to evaluating legal expert systems. Refs. [23,24] and the Seventh International Conference on Artificial Intelligence and Law [23] focus on this issue. In Section 6.1 we discuss how Split-Up has been evaluated.

6.1. Evaluating Split-Up

Stranieri [21] discusses how the Split-Up system has been evaluated in the following four distinct ways.

6.1.1. Domain expert evaluation of Split-Up

Our two domain experts assessed both the content and structure of the Split-Up knowledge base and the problem-solving strategy employed in Split-Up. The factor tree and argument structure used in the percentage split task were viewed positively by both domain experts associated with the project and four independent family law practitioners.

6.1.2. Lawyers evaluation of Split-Up

A comparison of predictions made by Split-Up with those made by eight lawyers on the facts from the same three cases was discussed in Ref. [41]. In two of the three cases, all eight lawyers agreed with each other (deviations of 5% either way from the Split-Up determination were deemed acceptable) and with the system. The third case presented significant controversy. Split-Up awarded the husband 55% of the assets. The lawyer's predictions varied from 20% to 60%. The four lawyers that produced outcomes that varied with the other lawyers and Split-Up assumed that the wife had contributed significantly more than the husband to the homemaker role. The case facts indicated that the household duties and child rearing was performed by hired helpers. We (the users of Split-Up) interpreted this as an equal contribution to the homemaker role whereas the four lawyers assumed the wife made the major contribution because the husband was fully occupied with his medical practice and was therefore unlikely to have the time to supervise household staff. This illustrates an important problem with the use of legal decision support systems—*users need to interpret data*. Many disputes are about interpreting data (or facts): for such problems human input is vital.

6.1.3. Performance of Split-Up on a new case

An in-depth evaluation on the use of Split-Up on a new trial case (recently concluded in the Family Court of Australia, namely, Opie vs. Opie) was conducted. The case is an unreported 1996 case tried by Justice Brown in the Melbourne registry of the Family Court of Australia (the cases used in the Split-Up system were taken from the Melbourne registry of the Family Court of Australia in the period 1992 through 1994). The marriage lasted 17 years and resulted in two children—of ages 14 and 16 at the time of the trial. The husband ran a business in the automotive industry that rarely returned large profits and no longer exists. The wife primarily worked as the homemaker but often worked part-time in the business. The 'common pool' system determined that the total assets for consideration was US$108,800. Both are in the mid-40s and of good health. The wife is to have custody of the children.

Split-Up determines the percentage split in terms of needs, contributions and the level of wealth of the marriage. For the case of Opie vs. Opie, Split-Up determined:

1. the marriage is considered to be less than average in wealth;
2. overall the husband has contributed the same as the wife during the course of the marriage;
3. in the future the husband's needs are less than those of the wife.

From these three determinations, through the use of a neural network, Split-Up determined Mr. Opie should receive 35% of the common pool. In her decision, Justice Brown granted Mr. Opie 34.7% of the common pool.

(1) was inferred through the use of a rule-based system given the value of the common pool. Domain experts claim the wealth of a marriage is important as future needs are significant for impoverished marriages but far less important for wealthy marriages, where each partner's needs will be met save for exceptional cases. With regard to contributions, as in (2), Split-Up suggested that the husband and wife contributed equally to the marriage. Justice Brown said that given the length of the marriage, the parties should be taken to have contributed equally.

With regards to (3), Split-Up suggested that the wife had greater future needs than the husband. The system came to this conclusion because it inferred that the wife's prospects for the future are not so fair —as she has poor future employment prospects and few resources. The husband, on the other hand, has fair future prospects, because he has good work prospects and some resources for the future. Justice Brown thought likewise.

We also compared Split-Up outputs with five written judgments of the Family Court of Australia. These cases were heard in 1995 and 1996 (the cases used in both the Split-Up training and test sets were decided in the 3 years between 1992 and 1994). This comparison showed that Split-Up inferences were similar to those decided by a judge. Many factors were left implicit in some judgments that Split-Up currently makes explicit. Some departures displayed by Split-Up from conclusions made in judgments can readily be made by small sample size.

6.1.4. Categorising Split-Up users

Current research on user evaluation involves obtaining feedback from users in four different categories: judges, registrars, mediators and lawyers. Members of each of these groups use Split-Up predictions and explanations. Our research is based on the work of Buchanan et al. [8], who claim that empirical validation with the use of a properly constructed questionnaire is a very useful quantitative indicator of user acceptance. Until now, financial resources to undertake such an evaluation have not been available. The attainment of the SPIRT grant mentioned previously will allow us to perform a thorough evaluation of the system.

We are using seven lawyers, four registrars, three judges and five lay people to evaluate the system using the quantitative assessment evaluation framework of Reich [19].

When first proposed, it was expected that the system would be primarily used by judges and lawyers. Our subsequent research [43] has shown our initial expectations as to who would be the main beneficiaries of the Split-Up system to be inaccurate.

6.1.4.1. How mediators use Split-Up. Mediators in family law input both parties facts, peruse the resultant prediction and then explore the hierarchy of relevant data, warrant and backing factors with the parties in order to inform and educate them. Points of convergence between the two parties become obvious and the scale and loci of compromise are more easily identified.

6.1.4.2. How lawyers use Split-Up. A lawyer uses the system a number of times with each client to explore hypothetical scenarios. Typical questions that arise are—*what difference in outcome is there if I argue that my client performed an equal share of the homemaker duties as opposed to arguing that she did most of those duties*? A consultation with the system offers a prediction in both scenarios and assists a lawyer in determining which argument to proceed with. Lawyers are less interested in exploring warrants and backings unless these relate precedents that will be used to substantiate an argument chosen.

6.1.4.3. How judges use Split-Up. Judges are required to arrive at an equitable outcome in the shortest amount of time possible. They have no need to educate litigants nor do they particularly need (or want) to evaluate their own judgments. However, they need to reach interim conclusions leading to a final judgment. They often need to interrupt a case for hours or days and then succinctly and quickly remind themselves of the facts and their own interim conclusions.

6.1.4.4. How divorcees use Split-Up. Divorcees with little knowledge of family law have often been surprised at predictions provided by the system. They tend to explore all warrants and backings in order to understand the prediction. Ultimately, it is not wise for systems such as Split-Up to be utilised by users with little family law knowledge, since such users cannot identify unusual (or hard) cases. The distinction between easy and cases may be jurisprudentially questionable, in that a case that seems perfectly commonplace today may be subsequently used to fundamentally alter a legal principle (hence becoming a landmark case). However, in practice, the distinction between commonplace and landmark cases is used, by the Family Court, on a daily basis, in order to decide which cases are to be published by court reporting services.

6.2. Evaluating IFDSSEA

Two schools in New York City School District Six were selected as test sites. The teachers selected to participate in the test were Grade 4 teachers. Training was provided so that the teachers would understand how to use the expert fuzzy classification scoring system.

All Grade 4 PAL tests completed by fourth grade students in these two schools were scored by teachers using the expert fuzzy classification scoring system. Over a 1-month period, 255 student writing samples were evaluated. At the end of the 1-month testing period, expert teacher graders from outside these schools reviewed the exams scored with IFDSSEA. They unanimously agreed the system was consistently scoring the essays according to the rules.

A controlled experiment was set up to determine just how effective teachers evaluating student writing samples with IFDSSEA were compared with domain experts (the teacher graders). Two hundred student writing samples were selected for the experiment. The three expert teacher graders reviewed each of the 200 writing samples and made an evaluation using the holistic scoring criteria. The same 200 were independently reviewed and assessed by three different teachers using the IFDSSEA. The results indicated that the teachers using IFDSSEA agreed with the three domain experts in 170 of the 200 cases for an agreement rate of 85%. Since it is not unusual for teachers to disagree on the score to be assigned to the same sample of student writing, most standardised writing exams allow a difference of one point between the two graders before a compromise must be reached. Using this criteria, 194 of the 200 cases could be considered in agreement (97%). In addition, the teachers using IFDSSEA scored the exams in one-third less time than the domain experts.

7. Conclusion

Until recently, the principle tools in the artificial intelligence arsenal were centred on symbol manipulation and predicate logic, while the use of numerical techniques were looked upon with disfavour. What is more obvious today is that symbol manipulation and predicate logic have serious limitations in dealing with real world problems in the realm of decision making. In this paper, we have focused on how soft computing techniques—in particular fuzzy logic and neural networks—can help build intelligent decision support systems.

The two examples we have considered—Split-Up and IFDSSEA—serve to demonstrate how statistical and symbolic techniques can be combined to provide for more effective decision making in knowledge-based systems. Whilst both systems use sophisticated domain techniques, they also allow for reasoning from imprecise data. With the growth of large databases, and the subsequent use of knowledge discovery and data mining techniques, the use of soft computing is essential. By integrating statistical and symbolic techniques, we are likely to see the development of more regular real world applications.

References

[1] K.D. Ashley, Case-based reasoning and its implications for legal expert systems, Artif. Intell. Law 1 (2) (1992) 113–208.

[2] E. Bellucci, J. Zeleznikow, Family negotiator: an intelligent decision support system for negotiation in Australian Family Law, Proceedings of the Fourth Conference of the International Society for Decision Support Systems, Lausanne, International Society for Decision Support Systems, Lausanne, Switzerland, 1997, pp. 359–373.

[3] E. Bellucci, J. Zeleznikow, A comparative study of negotiation decision support systems, Proceedings of the Thirty-First Hawaii International Conference on System Sciences, Los Alamitos, CA, IEEE Computer Society, Los Alamitos, CA, 1998, pp. 254–262.

[4] E. Bellucci, J. Zeleznikow, Artificial intelligence techniques for modeling legal negotiation, Proceedings of Seventh International Conference on Artificial Intelligence and Law, ACM, Oslo, 1999, pp. 108–116.

[5] T.J.M. Bench-Capon, D. Lowes, A.M. McEnery, Argument-based explanation of logic programs, Knowl.-Based Syst. 4 (3) (1991) 177–183.

[6] H.C. Black, Black's Law Dictionary, West, St. Paul, MN, 1990.

[7] J.M. Bonham, Cognitive mapping as a technique for supporting international negotiation, Theory Decision 34 (1993) 255–273.

[8] Buchanan, Moore, Forsythe, Carenini, Ohlsson, Banks, An intelligent interactive system for delivering individualised information to patients, Artif. Intell. Med. (1995) 117–154.

[9] J. Diederich, Explanation and artificial neural networks, Int. J. Man-Mach. Stud. 37 (1992).

[10] R.C. Eberhart, Using evolutionary computation tools in explanation facilities, Int. J. Expert Syst. 8 (3) (1995) 277–285.

[11] Fayyad, Piatetsky-Shapiro, Smyth, The KDD process for extracting useful knowledge from volumes of data, Commun. ACM 39 (11) (1996) 27–34.

[12] Ferns, LIFENET: tool for risk assessment of adolescent suicide, Expert Syst. Appl. 9 (2) (1995) 165–176.

[13] Fisher, Ury, Getting to YES: Negotiating Agreement Without Giving In, Houghton Mifflin, Boston, 1981.

[14] Gorry, Scott-Morton, A framework for management information systems, Sloan Manage. Rev. 13 (1) (1971).

[15] B. Kosko, Fuzzy Thinking: The New Science of Fuzzy Logic, Hyperion, New York, 1993.

[16] Nolan, A conceptual model for an intelligent fuzzy decision support system, Heuristics 10 (2) (1997) 31–43.

[17] Nolan, An intelligent system for case review and risk assessment in social services, AI Mag. 19 (1) (1998) 39–46.

[18] O'Leary, Goul, Moffitt, Radwan, Validating expert systems, IEEE Expert 5 (1990) 51–58.

[19] Reich, Measuring the value of knowledge, Int. J. Human-Comput. Stud. 42 (1) (1995) 3–30.

[20] Skabar, Stranieri, Zeleznikow, Using argumentation for the decomposition and classification of tasks for hybrid system development, Progress in Connectionist-Based Information Systems: Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems, Springer-Verlag, Singapore, 1997, pp. 814–818.

[21] Stranieri, Automating Legal Reasoning in Discretionary Domains, PhD Thesis, La Trobe University, Melbourne, Australia (1998).

[22] Stranieri, Zeleznikow, A re-examination of the concepts of open texture and stare decisis for data mining in discretionary domains, Proceedings of Eleventh International Conference on Legal Knowledge-based Systems, Koniklijke Vermand, Groningen, Netherlands, 1998, pp. 101–111.

[23] A. Stranieri, J. Zeleznikow, Evaluating legal expert systems, Proceedings of Seventh International Conference on Artificial Intelligence and Law, ACM, Oslo, 1999, pp. 18–24.

[24] A. Stranieri, J. Zeleznikow, Knowledge acquisition benefits of a non-conventional rule-based reasoning system for the

World Wide Web,IASTED International Conference Law and Technology, ACTA Press, Anaheim, CA, 1999, pp. 88–93.

[25] A. Stranieri, J. Zeleznikow, A survey of argumentation structures for intelligent decision support, Fifth International Conference of the International Society for Decision Support Systems, 1999.

[26] Stranieri, Gawler, Zeleznikow, Toulmin structures as a higher level abstraction for hybrid reasoning, Proceedings of the Seventh Australian Artificial Intelligence Congress, 1994, 203.

[27] A. Stranieri, P. Massey, J. Zeleznikow, Inferencing with legal knowledge represented as diagrams, in: A.W.F. Williams (Ed.), Poster Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence, University of New England, Armidale, Australia, 1994, pp. 25–32.

[28] Stranieri, Zeleznikow, Gawler, Lewis, A hybrid–neural approach to the automation of legal reasoning in the discretionary domain of Family Law in Australia, Artif. Intell. Law 7 (2–3) (1999) 153–183.

[29] Sycara, Machine learning for intelligent support of conflict resolution, Decis. Support Syst. 10 (1993) 121–136.

[30] Sycara, Multiagent systems, AI Mag. 19 (2) (1998) 79–92.

[31] Toulmin, The Uses of Argument, Cambridge Univ. Press, Cambridge, 1958.

[32] E. Turban, J. Aronson, Decision Support Systems and Intelligent Systems, Prentice-Hall, London, 1998.

[33] Weiss, Kulikowski, Computer Systems That Learn, Morgan-Kaufman, California, 1991.

[34] M.P. Wellman, Formulation of Tradeoffs in Planning Under Uncertainty, Pitman, London, 1990.

[35] M.J. Wick, W.B. Thompson, Reconstructive expert system explanation, Artif. Intell. 54 (1992) 33–70.

[36] J. Yearwood, A. Stranieri, The integration of retrieval, reasoning and drafting for refugee law: a third generation legal knowledge-based system, Proceedings of Seventh International Conference on Artificial Intelligence and Law, ACM, Oslo, 1999, pp. 117–126.

[37] Zadeh, Fuzzy logic, neural networks and soft computing, Commun. ACM 37 (3) (1994) 77–84.

[38] Zadeh, Kacprzyk (Eds.), Fuzzy Logic for the Management of Uncertainty, Wiley, New York, 1992.

[39] J. Zeleznikow, Building intelligent legal tools—the IKBALS project, J. Law Info. Sci. 2 (2) (1991) 165–184.

[40] J. Zeleznikow, D. Hunter, Building Intelligent Legal Information Systems: Knowledge Representation and Reasoning in Law, Kluwer Computer/Law Series 13, Deventer, Netherlands, 1994.

[41] Zeleznikow, Stranieri, Modelling discretion in the Split-Up system, The Pacific Asia Conference on Information Systems, Information Systems Research Management, Queensland University of Technology, 1997, pp. 307–320.

[42] Zeleznikow, Stranieri, Knowledge discovery in the Split-Up project, Proceedings of Sixth International Conference on Artificial Intelligence and Law, ACM, 1997, pp. 89–97.

[43] Zeleznikow, Stranieri, Split-Up: the use of an argument-based knowledge representation to meet expectations of different users for discretionary decision making, Proceedings of Tenth Annual Conference on Innovative Applications of Artificial Intelligence, AAAI/MIT Press, Cambridge, MA, 1998, pp. 1146–1151.

[44] J. Zeleznikow, G. Vossos, D. Hunter, The IKBALS project: multimodal reasoning in legal knowledge-based systems, Artif. Intell. Law 2 (3) (1994) 169–203.

[45] J. Zeleznikow, D. Hunter, A. Stranieri, Using cases to build intelligent decision support systems, in: R. Meersman, L. Mark (Eds.), Proceedings of the IFIP Working Group 2.6 Conference Stone Mountain, Georgia, USA, May 30–June 2, 1995, Chapman & Hall, London, UK, 1997, pp. 443–460.

John Zeleznikow is the Director of the Donald Berman Laboratory for Information Technnology and Law at the Applied Computing Research Institute, La Trobe University. His research interests are in the areas of intelligent decision support systems, knowledge discovery from databases, and hybrid reasoning. He has published two books on artificial intelligence and law.

James Nolan is a professor in the Quantitative Business Analysis and Computer Science departments at Siena College. He holds a PhD from the Rockefeller College at the University of Albany. His research interests are in the areas of statistical and machine learning, hybrid intelligent systems design and development, fuzzy decision support systems, and neural networks. He has published numerous articles in these areas.