Using soft computing to model discretionary decision-making in the Split Up system

John Zeleznikow

Database Research Laboratory, Applied Computing Research Institute
La Trobe University
Bundoora, Victoria, Australia, 3083
johnz@latcs1.cs.latrobe.edu.au

Andrew Stranieri

School of Information Technology and
Mathematical Sciences
University of Ballarat, PO Box 663
Ballarat, Victoria, Australia, 3353
a.stranieri@ballarat.edu.au

Abstract

There are many decision support systems to aid decision making in structured domains. Such domains are characterised by involving repetitive and routine tasks. They involve definite procedures for supporting decision making. Far fewer tools are available for reasoning in discretionary domains. Nevertheless, most management decision making is indeed unstructured — the decision maker must provide judgement, evaluation and insights into the problem definition. In this paper we focus upon decision making in a discretionary domain — namely how Australian judges distribute marital property upon divorce. Our system, Split Up provides such advice. The system has been developed in the object oriented expert system tool KnowledgePro.

Previously, we described a methodology for measuring the degree of discretion judicial decision makers are given. The important features in determining the amount of discretion in a legal domain are the number and nature of open textured and bounded predicates.

The task of determining what property a Family Court judge may distribute is determined to be narrow and bounded. A rule—based system (using directed graphs) has been developed to perform this task.

The task of deciding what percentage of the Common Pool the husband is to receive is deemed to be wide and bounded. A hybrid rule-based/neural network provides advice in this domain.

Neural networks provide no explanation for their answers. Critical legal theorists claim judges reach their decisions based on their value systems. In their written judgement, they provide a rationalisation to support the decision(s) they have made. In a similar manner our system provides a rationalisation for the conclusion reached using. The explanation is reached through using the argumentation theory of Toulmin.

Introduction

Stranieri et al (1997) concluded that the important features for modelling legal domains are the extent to which a task is both open textured and bounded. Open textured legal predicates contain questions that cannot be structured in the form of production rules or logical propositions and which require some legal knowledge on the part of the user in order to answer. A domain may be said to be 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. We concluded that legal domains could be divided into four quadrants depending upon their degree of boundedness and open texture.

Gorry and Scott-Morton (1971) define unstructured decisions as those in which the decision maker must provide judgement, 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 (1994) argue that one can used rule-based systems to model structured domains.

In November 1991, Graham, J. of the Melbourne registry of the Family Court of Australia asked us to build a knowledge based system to model that part of the Family Law Act (1975) which dealt with the distribution of assets upon the dissolution of a marriage. We were originally reluctant to become involved in such a venture because we felt the Act was too discretionary to be modelled. Indeed, this is true for that part of the Act which deals with the welfare of children. This is because the under section 64(1) (a) *the court must regard the* *welfare of the child as the paramount consideration.* Neither parliament nor the courts have clearly defined what are the paramount interests of the child, and so we viewed this as an extremely open textured term.

However, we noted that there are established mechanisms for determining the distribution of marital property upon divorce. In this paper, we discuss the development of the Split Up system which offers advice upon the distribution of marital property upon divorce. Our system uses an integrated rule based system/neural network to advise upon how the property is distributed. Zeleznikow and Stranieri (1997) have shown the use of neural networks and soft computing is particularly useful when discovering knowledge from domains which have an abundance of commonplace cases. Commonplace cases are those cases which are unreported and do not feature as precedents for future decision-making. As individual cases, commonplace cases have minimal influence, they are only significant when grouped together with a large amount of other cases. Landmark cases provide a normative structure for subsequent reasoning and are prominently used in legal decision making. When designing the Split-Up system we focussed upon the use of commonplace cases.

Decision making in Australian Family Law

Few legal reasoning systems have been developed in discretionary domains. Edwards and Huntley (1992) applied rule—based reasoning to the discretionary domain of Family Law in Scotland and reported some inadequacies of that approach. Following the original request of Graham, J. we first built a rule—based system which advised upon marital property division upon divorce. This was reported by Stranieri and Zeleznikow (1992). Whilst this system gave us reasonably accurate answers, it failed to take into account either legal theory or the notion of discretion.

Our research is based upon the theory of legal realism — that judges make decisions for a range of reasons which cannot be articulated or at least are not apparent on the face of the judgement written. According to legal realists it is meaningless to argue that a judicial decision is made according to any existing rules, rather a decision is more a reflection of the judge's biases. As Zeleznikow and Hunter (1994) note, once the judge has made the decision based on these biases then she will find a legal rule on which to justify the decision, ex post facto.

Ingleby (1993) notes that Australian family law allows its judges much discretion in decision making. Until 1990, Family Court judges were not required to give reasons as to why they reached their decision. Despite the requirement for Family Court judges to justify their decision-making, we would argue that the manner in which judges reach their decisions can be quite different from the manner in which they justify them. To model judicial decision making with respect to the division of property under Australian Family Law we needed to discover the relative significance of each of those issues which judges use to distribute marital property.

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 the 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 common pool determination task was suited to a rule based reasoning approach although the reasoning process for this task is not explicitly governed by any statute. The elicitation of expert heuristics for a rule based system was initiated using a structured interview technique with a domain expert, Renata Alexander who has over twenty years experience with the Legal Aid Commission of Victoria.

The creation of if-then rules from transcripts of structured interviews proved to be a time consuming and cumbersome process. In order to attempt to accelerate this process, we encouraged the expert to represent her own dialogue with a hypothetical client as a directed graph. In this way, fifty one graphs containing two hundred and thirty nodes were elicited in thirteen, one hour sessions.

However, this approach of itself is insufficient when attempting to deal with open texture and discretion. Having completed the common pool determination, the judge then determines what percentage of the common pool each partner is to receive. A rule based approach is inappropriate for constructing this module. The section of the Act dealing with the percentage of the common pool each partner receives is highly discretionary. This is because the Family Law Act (1975) lists a number of factors to be considered for a percentage split determination yet provides no guidance on the relative significance of each factor or on how they are to be combined.

This domain is considered discretionary because of what is described as a 'shopping list' of relevant factors. Different judges may, and do, reach different conclusions based on the same facts, since each judge assigns different relative weights to each factor. Ascertaining knowledge about how a judge weights and combine factors is difficult in that a guessed numerical weighting is unlikely to represent the actual weight of the factor in the context of a large number of interdependent factors.

Although the statute presents a flat list of relevant factors without specifying how these factors relate to each other, we believe domain specific knowledge is crucial in specifying relationships between factors. Domain expertise is also critical for the elicitation of the

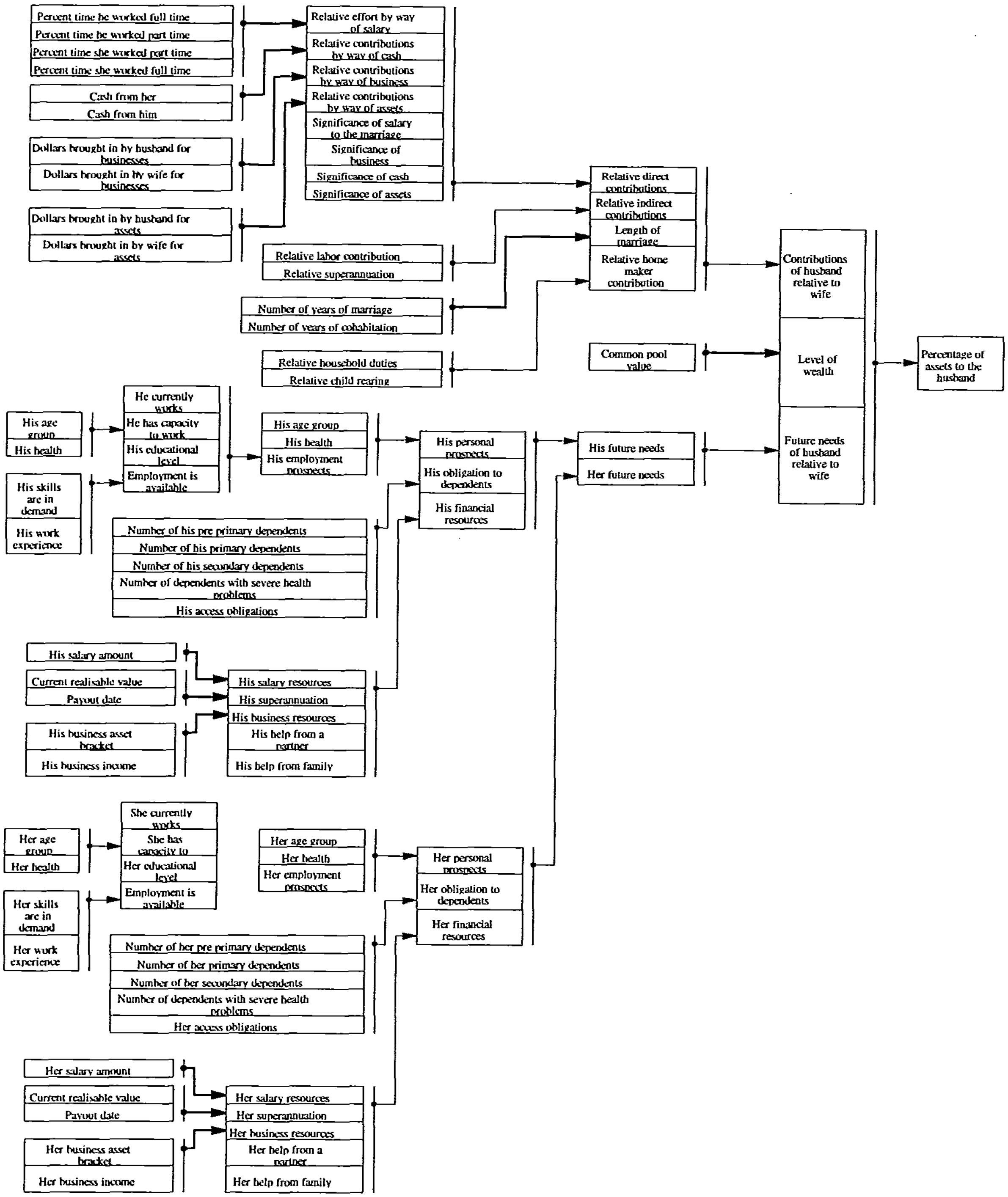

Figure 1. Hierarchy of factors for percentage split determination

other factors which are relevant but not explicitly mentioned in the statute. Figure 1 illustrates ninety nine factors which domain experts indicated were relevant for a percentage split prediction. These factors are placed in a hierarchy by experts though no attempt was made at this stage to elicit the way in which factors combine. The way some factors combine was later learnt by machine learning algorithms known as neural networks. The way other factors combined was modelled with rules that derived from expert heuristics.

The arguments in figure 1 are inferenced by either rules ('solid line') or neural networks ('broken line'). The choice of inferencing mechanism chosen depended upon the open texturedness and boundedness of the factor. Figure 1 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 contributions, relative indirect contributions, the length of the marriage and the relative contributions of both parties to the home-making role.

No attempt is made in figure 1 to represent the way in which relevant factors combine to infer factors higher in the hierarchy.

The Split Up system has been heavily tested by judges, judicial registrars, mediators, counsellors, legal aid lawyers and private practitioners expert in the domain of family law. As well as being impressed by the system, most have stressed that the 'hierarchy of relevant factors for percentage split determination' provides an incredible source of knowledge as to how a systematic decision with regard to the distribution of marital property, can be made.

Using neural networks to model discretionary decision making

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 which 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. For our purpose a neural network will be considered as a pattern matching statistical technique for learning the weights of each of the relevant attributes used in the decision making process.

Neural networks determine and represent weights of factors sub-symbolically and thus are well suited to capturing the weighting of factors which predict a judge's performance. They were thus our preferred option for performing knowledge discovery amongst our Australian Family Law cases. A detailed description of how neural networks are used in the Spit—Up project can be found in Zeleznikow and Stranieri (1996).

Factors specified by the statute as relevant for a percentage split determination can be selected as inputs into a neural network and the output can be the percentage of the assets awarded to the parties. A supervised network was preferred over an unsupervised network as the output, namely the percentage split reported in a judgment, is known in all cases.

A pool of four hundred unreported cases was used as an initial source for the extraction of case facts so that a training set may be assembled. Many of the cases available to us proved to be unsuitable because they involved a custody decision in addition to a property determination. Data was extracted manually from one hundred and three cases that revolved exclusively around a property determination. The extraction of variable values from case judgments was performed by two raters. Inter rater comparisons were performed at random intervals by having both raters extract data on the same case.

The hierarchy of figure 1 can be seen to provide a framework for the decomposition of the task of predicting a percentage split into thirty five sub tasks. Outputs of tasks further down the hierarchy are used as inputs into tasks further along the hierarchy. In Split—Up, outputs in twenty-one tasks were inferred from their respective inputs with the use of neural networks. The remaining task outputs were inferred with the use of rule sets.

Figure 2 illustrates the framework for inferring a percentage split outcome with the use of a neural network. The inputs to the neural network are values on each of the three relevant factors, contributions, future needs and wealth. The neural network's output is the percentage split predicted.

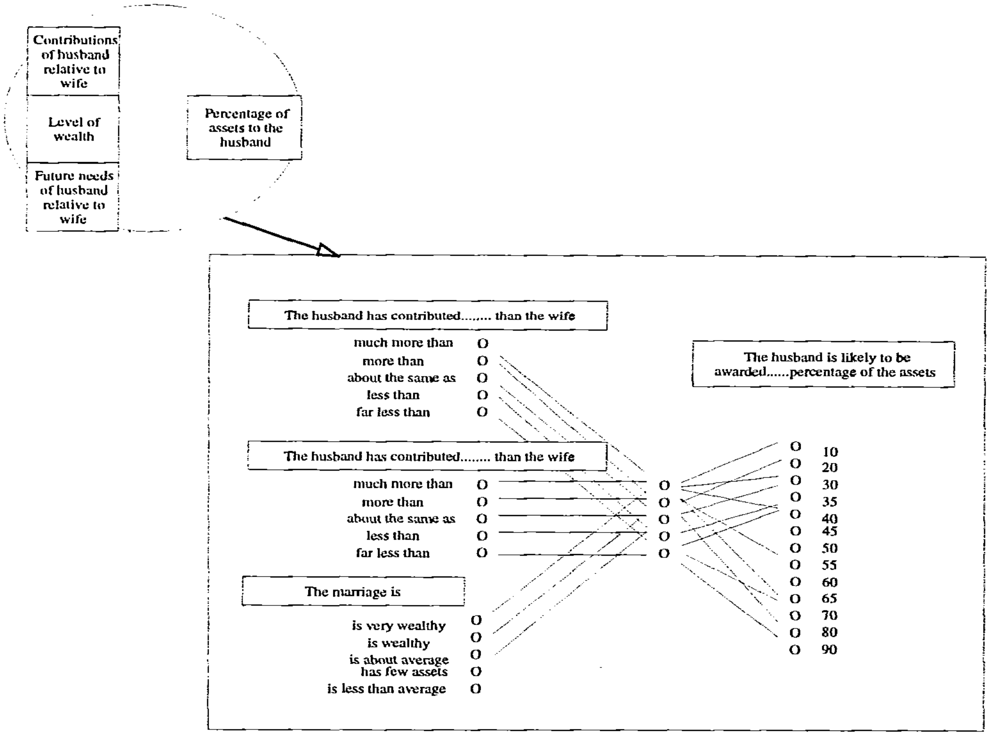

Figure 2 — Neural network for percentage split determination

Three fundamental difficulties are apparent in our approach:

• how to ascertain which features of a case to extract;

• how to glean anything of worth from a small data set;

• how to provide explanations for neural network outputs. In law a basis for the expert outcome is vital — lawyers are hardly likely to accept the output of a knowledge discovery algorithm without further justification.

Our approach to determining a case template was to use domain experts to advise about which factors to include. Thus even though our project involves automated knowledge discovery (learning the weights of certain attributes through the use of neural networks) we made extensive use of domain expertise. By subdividing the task of percentage split determination into a sequence of smaller sub tasks (see figure 1) we managed to construct an intelligent system using one hundred and fifty cases.

Providing explanations in the Split Up system

Despite having judicial discretion Australian Family Court judges are required to justify their decisions. We believe that the use of Toulmin argument structures can provide adequate explanations for advice proffered from a knowledge discovery algorithm.

Toulmin (1958) concluded that all arguments consist of four invariants: claim, data, warrant and backing. 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.

Figure 3 provides an example of two Toulmin argument structures in the Split Up system. The claim of the argument on the left serves as data for the argument on the right.

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 thirty five sub-tasks where each sub task represents an argument. Many of these arguments produced claims which were in turn used as data for other arguments. All arguments contribute to a culminating argument— the percentage split illustrated on the right of figure 1.

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 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 which 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.

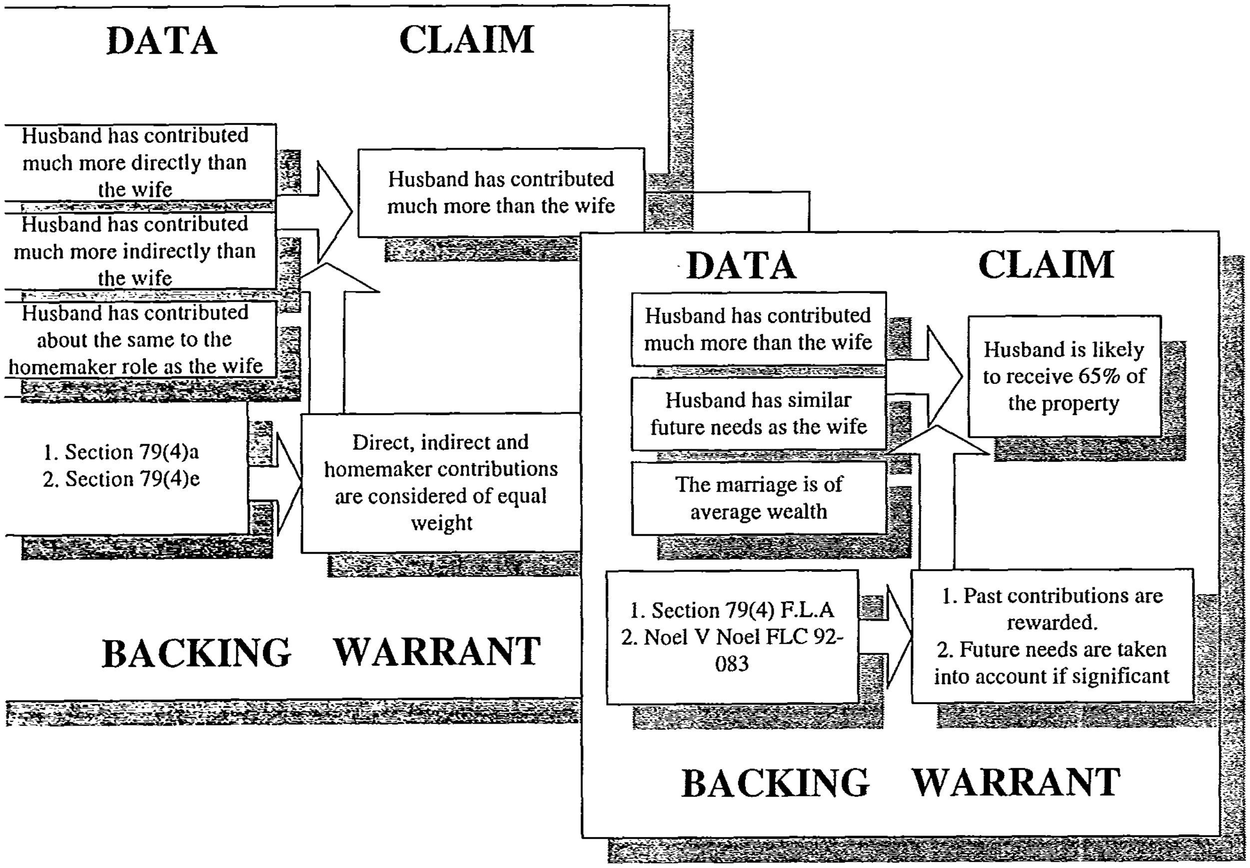

Figure 3 — Two Toulmin arguments in Split—Up

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, Knowledge Pro. The hypertext facilities built into Knowledge Pro allow the warrant and backing based explanations to draw on statutes and past cases. Those arguments which 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.

Data mining in the Split Up system

The data mining phase in Split—Up was performed with neural networks. All networks were trained using 5 fold cross validation. Cross validation is a resampling technique described by [Weiss and Kulikowski 1992] that provides a mechanism for estimating the error rate of a classifier on the true population. The simplest way to define a classifier error is to count the number of examples correctly classified by a neural network.

Counting the number of correctly classified examples leads to a measure of network performance which may be too fine-grained for legal applications. A better measure of a network's performance includes an indication of the magnitude of the error. A variation of 5% either way from a judge's decision of the percentage of assets awarded to the wife, is, in our view, a minor error. A network which outputs a percentage split which deviates from that obtained by a judge by 20% is assumed to have erred. Although the cut off point for declaring that an error has occurred is necessarily subjective, it was important that a metric be discerned which could be applied consistently to all networks in Split—Up.

We measure a network's performance by recording the position of the set bit in the neural output and compare this with the set bit in the expected output. This is possible in Split—Up networks because all inputs and outputs are binary. Consider a network with 5 binary outputs as an example. A network output of [1 0 0 0 0] for a particular example indicates that the first bit is set. If the actual output has the fifth bit set [0 0 0 0 1], we consider the network to have made an error of magnitude 4. If the actual output sets two bits such as it would in [0 1 1 0 0] we take the average of the positions of the set bits. In this case we say the actual set bit is in position 2.5. If the expected output was [0 0 1 0 0] then the error of magnitude is 0.5 (namely 3 - 2.5).

The error heuristic we used is central to the training of networks in Split—Up. Training is halted once the proportion of errors of magnitude 3 or more is observed to be 3% or less. An error of magnitude 3 represents a significant error for most networks. However, the cost of eliminating these errors totally is high in that the additional training required increases the risk of overtraining. Overtraining occurs when a classifier has been exposed to training data too many times, The classifier performs very well on training data but does not perform well on unseen examples. The classifier is said to have 'overfitted' which results in poor generalisation. An extreme error on only 3% of cases in the true population represents a margin that we considered tolerable in practice. Table 1 illustrates the topology and performance of a sample of networks in Split—Up.

Conclusion and Future Research

In the Split Up system we have considered modelling judicial decision-making in the domain of Australian Family Law. Family law varies from other legal domains is that in general:

1. There are no winners or losers — in most common law domains one party to a legal dispute wins a case whilst the other loses. In civil matters, under the cost indemnity rule, the loser of a litigated case plays the costs of the winner. Admittedly in child welfare matters, it might be the case that one parent is given sole custody — but save for exceptional circumstances the other parent will be given access and joint guardianship.

Thus Family Law negotiation and decision making has more in common with management decision making than do most legal domains.

2. There are a vast amount of litigated Family law cases each year — In Australia there are approximately 100,000 divorces each year, of which 5,000 cases are litigated and 1,000 go to judgement. In most other civil domains[1] the number of litigated cases which go to judgement are less than a score. Thus the use of neural networks, or indeed many other soft computing techniques, to model discretionary decision-making in Australian Family law is at least feasible.

3. Parties to a family law case often need to communicate after the litigation has concluded. Hence the Family Court encourages negotiation rather than litigation.

Fisher and Ury (1981) developed the concept of Principled Negotiation. This concept promotes deciding issues on their merits rather than through a haggling process focussed on what each side says it will and will not do. A major feature of Principled Negotiation is to know your best alternative to a negotiated agreement (BATNA) since the reason you negotiate with someone is to produce better results than would otherwise occur. In Bellucci and Zeleznikow (1996), we indicate how the Split Up system can be used to determine the BATNA for each party to a family law dispute.

We have concluded that we can model judicial discretion-making through the use of a hybrid rule-based/neural network system. The major drawback of such a system is however its inability to provide explanations.

[1] Australia does not allow plea bargaining in criminal law cases. Hence, in contrast to the United States, many criminal law disputes are litigated. We have been reluctant to model discretion in criminal law, since as with child welfare determinations, it is an unbounded domain — lawyers have great flexibility in the issues which they can raise in court.

Network name	Topology: Input-Hidden-Output	Average proportion of errors of magnitude > 3	Average proportion of errors of magnitude > 2	Average proportion of errors of magnitude > 1	Average proportion of errors of magnitude > 0.5	Number of epochs
Percentage Split	15-12-13	0.03	0.12	0.16	0.31	900
Relative contributions	20-8-5	0.01	0.02	0.06	0.27	1130
Relative needs	8-3-5	0.00	0.01	0.02	0.07	230
Individual needs	14-3-4	0.02	0.07	0.11	0.30	830
Individual personal prospects	17-9-5	0.02	0.01	0.05	0.26	480
Individual employment prospects	14-8-5	0.02	0.02	0.08	0.10	1170
Individual capacity to work	12-5-3	0.01	0.00	0.06	0.22	180

Table 1 — Performance of Split—Up neural networks

Since our research is based upon the theory of legal realism, we use the argumentation theory of Toulmin to provide explanations which are independent of the technique used to provide the original determination.

Problems we are currently addressing include:

(1) **Selecting appropriate attributes for use in the Split-Up system** — the current version of Split-Up asks questions about ninety-nine attributes. We are currently in the process of applying feature selection methods using genetic algorithms to search for optimal decision trees that are induced using subsets of features (Skabar, Stranieri and Zeleznikow, 1997).

(2) **Concept drift** — how do cases vary over both regions and time. Schlimmer (1987) defines concept drift as the change in concepts over time. The commonplace cases used in Split—Up's training set were decided by judges in 1993. At that time judges were required to assess the contributions of each partner. Since then there may have been a considerable change in judges' attitudes to determining past contributions. There have indeed been calls from politicians to amend the Family Law Act to imply a presumption that both parties to a marriage contributed equally. Further a departure from this presumption would occur only in exceptional circumstances.

It should be noted that most of the cases used in the Split Up training set come from the Southern registry of the Family Court of Australia. This is because data from the other registries was unsuitable for inclusion in our system. However this means that the Split Up system models the thinking of judges in Melbourne, rather than the more conservative outlying states. Feedback from users of the system appears to stress this fact. Thus we need to consider concept drift over both time and location.

(3) **Comparing negotiated and tried cases** — we are currently using the Split Up system and negotiated cases supplied to us by twelve family lawyers who are trialling the Split-Up system to determine if there is a major difference in outcomes between negotiated and litigated cases.

(4) **Alternative soft computing algorithms** — in the Split-Up system we have used neural networks as our data mining technique. There is no reason why we could not have used regression analysis or indeed a variety of other statistical and soft computing techniques. We are however reluctant to use Bayesian classifiers or fuzzy computing due to the difficulty of ascribing probabilities to the outcomes of our commonplace cases.

(5) **Building computer tools to support human negotiation.**

References

1. Bellucci, E. and Zeleznikow, J. 1996. Family—Negotiator: an intelligent tool for supporting human negotiation. *Research and Development in Expert Systems XIII. Proceedings of Expert Systems 96*, Cambridge, SGES Publications:289-300.

2. Edwards, L. and Huntley, A. J. K. 1992. Creating a Civil Jurisdiction Adviser. Law, Computers and Artificial Intelligence: 1(1), 5—40.

3. Fisher, R. and Ury, W. 1981. *Getting to YES: Negotiating Agreement Without Giving In.* Boston: Haughton Mifflin.

4. Gorry, G. A. and Scott-Morton, M. S. 1971. A Framework for Management Information Systems. *Sloan Management Review*, 13 (1).

5. Ingleby, R., 1993. *Family Law and Society*, Butterworths, Sydney.

6. Schlimmer, J. C. 1987. Concept acquisition through representational adjustment. Ph.D. diss., University of California, Irvine, California.

7. Skabar, A., Stranieri, A. and Zeleznikow, J. 1997. Using argumentation for the decomposition and classification of tasks for hybrid system development. To appear in ICONIP'97: *International Conference on Neural Information Processing.*

8. Stranieri, A. and Zeleznikow, J. 1992. SPLIT-UP—Expert system to determine spousal property distribution on litigation in the Family Court of Australia. In the Proceedings of Artificial Intelligence Conference (Australia)-92, 51-56, Hobart: World Scientific.

9. Stranieri, A., Zeleznikow, J., Gawler, M. and Lewis, B. A. 1997. Split—Up: A hybrid rule—neural approach for the automation of legal reasoning in the discretionary domain of Australian family law. To appear in *Artificial Intelligence and Law.*

10. Toulmin, S. 1958. *The uses of argument.* Cambridge, Cambridge University Press.

11. Weiss, S. and Kulikowski, C. 1992. *Computer Systems that Learn: classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems.* Morgan Kaufman.

12. Zeleznikow, J. and Hunter, D., 1994. *Building Intelligent Legal Information Systems: Knowledge Representation and Reasoning in Law*, Kluwer Computer/Law Series, 13.

13. Zeleznikow, J. and Stranieri, A. 1996. Discovering knowledge in the Legal Domain: The Split Up system. Proceedings of Expert Systems'96 (ES'96). *Sixteenth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems.* SGES Publications. Cambridge. UK. pp271-288.

14. Zeleznikow, J. and Stranieri, A. 1997. Knowledge Discovery in the Split-Up Project. *Proceedings of Sixth International Conference on Artificial Intelligence and Law*, ACM: 89-97