Decision Making in Complex, Uncertain Domains: A Role for the 'Not So Expert' System?

G. Michael McGrath*

Centre for Services and Tourism Research

Victoria University

Melbourne, Australia

Email: michael.mcgrath@vu.edu.au

Joze Kuzic

Caulfield School of Information Technology

Monash University

Melbourne, Australia

joze.kuzic@infotech.monash.edu.au

Decision Making in Complex, Uncertain Domains: A Role for the 'Not So Expert' System?

Abstract

After falling out of favour during the 1990s, advisory expert systems and their underlying artificial intelligence (AI) technologies are now being used increasingly in organizational decision making activities – particularly, data mining. In this paper, a number of popular AI technologies are evaluated with respect to their application in greyhound racing tipping. Parallels between this domain and organizational forecasting activities are drawn. Early indications are that the forecasts produced by these technologies compare favourably with those of both human experts and previous, relevant decision support systems (DSS). Particular emphasis is placed on reducing complexity where possible – hence, the 'not so expert' system.

Keywords and phrases: forecasting; artificial intelligence; decision support systems.

INTRODUCTION

After generating a great deal of excitement during the 1980s, expert systems (and their associated artificial intelligence (AI) technologies) failed to deliver much in the way of anticipated benefits and, consequently, fell out of favour in many organizations (Luger, 2005; Beemer and Gregg, 2008). A fairly common perception – that expert systems had been entirely discredited and had no place in serious organizational IT activity – was, however, far from the truth: in particular, underlying AI technologies (such as fuzzy logic and neural networks) were gradually integrated into mainstream business applications and, in particular, the advent of data mining (Berson et al., 1999) as a key decision support tool provided many of these technologies with a new lease of life.

Major problems confronting early expert system developers were that development tools were relatively immature and that the computing hardware of that period was insufficiently powerful to support those tools in non-trivial, company-critical applications[1]. The extraordinary advances in both hardware and software over the past 20 years or so have largely eliminated these as relevant problems. In addition, as noted by Beemer and Gregg (2008), the evolution of 'expert' systems into 'advisory' systems appears to have resulted in a much greater level of technology acceptance within organizations.

Our purpose in this study is to evaluate the usefulness of modern AI technology in complex and uncertain domains. A related objective is to assess whether the 'not so expert' concept[2] (proposed by Debenham, 1985) might be employed to advantage (particularly with respect to reduced development and operational costs). The problem domain chosen was greyhound racing: principally because outputs

---

[1] For accounts of the types of problems encountered with these early applications, see (Debenham and McGrath, 1983; McGrath and Debenham, 1984).

are *absolute* (i.e. a dog either wins or it doesn't and pays one – and only one – amount in the event that it does win) and *timely* (i.e. one does not have to wait months or years to assess outcomes). While, at first glance, this problem domain might appear to have little to do with organizational life, several researchers (see, for example, Asch et al., 1984) have argued that there are strong similarities with racing prediction and some important organizational functions – particularly, various financial forecasting activities.

The paper is organized as follows: in the following section, some background on expert systems and the chosen problem domain is presented. This is then followed by sections dealing with our research and database designs respectively. Preliminary data analysis results are then presented and this is followed by a section detailing results of the application of betting guidelines induced during the initial data analysis phase. The final section contains a discussion of results and concluding remarks.

BACKGROUND: EXPERT SYSTEMS AND THE PROBLEM DOMAIN

Expert Systems: Evolution

Rule-based expert systems have been around for a very-long time; since the 1970s at least, with early examples including: *DENDRAL,* designed to infer plausible chemical structures from mass spectrographic data (Buchanan and Feigenbaum, 1978); *MYCIN,* a system for the diagnosis and treatment of infectious blood diseases (Shortcliffe, 1976); and *R1,* devised for the automatic configuration of Digital's VAX range of computers (McDermott, 1980). In turn these early systems were built upon foundational AI research in areas such as (declarative) logic programming (Kowalski, 1979), frame-based knowledge representation (Minsky, 1975), semantic networks (Deliyanni and Kowalski, 1979), deductive databases (van Emden, 1978) and intelligent inferencing techniques (Pereira and Porto, 1982).

Early expectations for expert systems were very high. For example, there was a fairly-common belief that, by capturing experts' knowledge in troubleshooting systems such as CATS-1[3], organizations could protect themselves from the loss of essential staff, could improve performance and reliability and perhaps, most importantly, could substantially reduce labour costs (Bonnistone and Johnson, 1983). Perhaps, all these early hopes and expectations were captured most spectacularly in the massive, billion-dollar, Japanese *5th Generation Computer Systems* project (Feigenbaum and McCorduck, 1983).

---

[2] Essentially, the 'not so expert' system is one that conforms to the 80:20 rule – whereby 80% of the functionality can be developed with 20% of the effort.

[3] A pioneering expert system designed for diagnosing and remedying problems with diesel-electric locomotive engines.

Despite some notable successes, however, many of these early expert system projects were perceived as failures. Common problems identified by Luger (2005) included: i) difficulties in capturing 'deep' problem domain knowledge; ii) a lack of robustness and flexibility; iii) poor explanations of how solutions were arrived at; iv) limited means of solution verification; and v) little learning from experience.

Beemer and Gregg (2008) suggest that (partly as a response to this) expert systems have now evolved into *advisory* systems. They distinguish this newer breed of intelligent systems by: their ability to cope with unstructured knowledge and uncertainty; the use of case-based reasoning (CBR) in lieu of the more traditional rule-based approach; support for iterative decision-making and environment monitoring; and the ability to cope more effectively with problem context. In particular, they claim that, whereas expert systems tend to focus on a narrow domain and present the user with a recommended course of action, advisory systems gradually (i.e. iteratively) guide the user towards a range of acceptable options and leave the ultimate decision up to the end-user.

While one might accept this prescription-advice distinction, we are inclined to the view that the other advisory systems distinguishing features detailed above apply just as much to expert systems. Moreover, differentiating the two system categories in this way tends to downplay the significance of many important, early AI research contributions and, perhaps more unfortunately, reinforce the view (all too common in the IT industry) that for something to be worthwhile it must be recent[4].

Nevertheless, it is now generally accepted that applications which satisfy Beemer and Gregg's (op. cit) criteria for advisory systems may lay a greater claim to the label, 'intelligent', than those that do not. Consequently, over the past 20 years or so, considerable advances have been made in fields such as fuzzy logic (Cox and O'Hagan, 1999) neural networks (Rojas, 1996) and various approaches to automated learning (Segaran, 2007). In particular, many of these methods and technologies have found their way into the various data mining (Berson et al., 1999) software packages now used by organizations to gain competitive advantage through business intelligence (BI) (Davenport and Harris, 2007).

One interesting result of the integration of these technologies into expert systems, however, is that more recent applications have become increasingly sophisticated and complex. A downside to this, of course, is that these systems are more costly to build and maintain. Over 20 years ago, however, Debenham (1985) proposed the idea of the 'not so expert' system, a concept which was, effectively, an instance of

[4] With reference to Beemer and Gregg's (op. cit) distinctions, it is worth noting that (for example) Ross Quinlan's pioneering work on reasoning with uncertainty was first published in 1983 and foundational work on CBR by Schank, Minsky and others dates back to the early-1970s (Minsky, 1975; Rieger, 1975).

the 80:20 applications development rule - where 80% of the functionality might be developed with only 20% of the total effort required (Abdel-Hamid and Madnick, 1991: 197-202).

This, essentially, is the approach we have taken here. Specifically, we have simplified a problem domain containing over 30 relevant variables to one with less than 10. We have, however, made extensive use of the technological advances discussed above; in particular, in the areas of rule-based development, data mining, CBR and reasoning with uncertainty.

Problem Domain

The domain chosen was greyhound racing and, more specifically, meetings conducted in the state of Victoria, Australia. The major objective of the system is to predict race winners and place-getters. Some of the key variables and dependencies between them are indicated in the causal-loop diagram (CLD) presented in Figure 1[5].

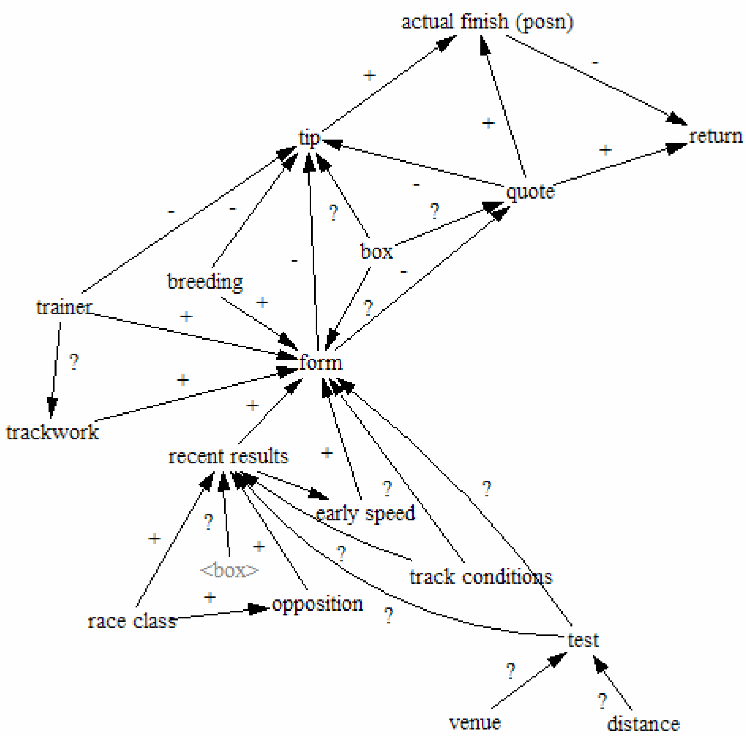

Figure 1: Greyhound racing: some key variables and dependencies.

Over the years, much research has been conducted into intelligent forecasting and decision support systems (DSS) in finance (for recent examples, see: Kim, 2004; Lee, 2004; and Tsang et al., 2004).

[5] Note that in CLDs a '+' annotation indicates that the two variables (i.e. the cause and effect variables at either end of an arrow) move in the same direction, while a '-' annotation means that they move in opposite directions. Here, some care should be taken with *tip, posn* and *quote* variables as, generally speaking, the lower these are, the better.

Several researchers (for example Asch et al., 1984; Hausch and Ziemba, 1985; and de la Maza, 1989) have noted very strong similarities between financial and horse racing forecasting and, indeed, Tsang and his colleagues (Tsang et al., 1998; and Tsang et al., 2004) have applied their *EDDIE* system (based on genetic algorithms) to both domains with considerable success. Much less attention, however, has been devoted to applications of intelligent systems in greyhound racing per se (for an exception, see Chen et al., 1994).

Obviously though, there are very strong parallels between horse and greyhound racing: for example, form, odds and experts' tips are very important when making selections; there is exactly the same trade-off between odds (quotes) and returns; and breeding, trainers' records, track conditions and the match between the object race and a dog's record over that test (venue and distance) are all vital factors in predicting outcomes. However, there are significant differences between the two domains: specifically: i) the greyhound racing domain seems to be somewhat more predictable; ii) win and place returns are generally lower with the dogs; iii) with eight runners in every race, greyhound racing is more standardized; and, perhaps, most importantly iv) a dog's starting box appears to be very important. With respect to this last point, Table 1 details the performance of each starting box, for each test considered in our study[6].

Table 1: Box performance by track and distance in the 12 months to 30/4/2008.

	Box Number									
30/04/2008	**Metres**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Warrnambool	332	17.7	14.3	13.1	13.1	10.9	5.7	8.0	17.1	100
Warrnambool	450	17.3	12.0	11.8	11.6	8.1	11.8	13.4	14.0	100
Meadows	525	24.3	16.0	11.5	11.0	8.4	8.4	10.3	10.0	100
Ballarat	450	19.4	12.2	9.0	8.3	11.8	10.1	11.1	18.1	100
Ballarat	550	26.2	14.2	12.0	12.0	11.5	6.6	9.8	7.7	100
Geelong	347	19.4	17.1	12.3	10.5	8.8	10.3	9.7	12.0	100
Geelong	457	15.6	11.3	11.3	9.0	12.4	8.9	12.9	18.7	100
Wangaratta	403	20.4	18.7	11.6	10.2	5.4	8.5	8.2	17.0	100
Wangaratta	474	18.1	14.1	10.5	8.1	13.3	7.7	9.7	18.5	100
Traralgon	298	27.5	12.1	12.8	10.7	9.4	6.7	10.1	10.7	100
Traralgon	513	22.4	14.1	9.0	11.6	10.8	8.2	7.7	16.2	100
Sandown	515	19.1	15.2	13.2	10.0	9.1	8.2	12.0	13.2	100
Cranbourne	311	25.3	13.4	6.5	6.5	10.2	7.5	8.6	22.0	100
Cranbourne	520	13.1	12.4	8.6	11.2	10.5	11.7	12.6	19.8	100
Shepparton	390	20.0	10.2	8.9	6.9	14.1	10.8	12.8	16.4	100
Shepparton	450	14.1	11.4	13.7	11.1	11.8	10.5	12.7	14.7	100
Sale	440	18.6	12.0	11.8	9.2	10.9	10.3	8.7	18.6	100
Sale	520	23.6	12.6	11.0	7.1	12.6	8.7	11.8	12.6	100
Bendigo	430	17.6	11.1	10.3	7.8	10.7	11.8	14.5	16.3	100
Bendigo	545	22.8	19.3	13.8	6.9	13.1	7.6	4.8	11.7	100
		20.0	13.7	11.0	10.1	10.4	8.8	10.6	15.4	

A brief glance at Table 1 seems to indicate that, for most tests, dogs that draw inside boxes have a distinct advantage, dogs racing from box 8 also seem to do quite well and dogs in the centre appear to be at something of a disadvantage. It seems reasonable to speculate that dogs racing from the inside have to travel less distance and, perhaps, dogs that draw on the extreme outside may get a 'sit'[7]. Whatever the reason though, it does seem that box position is, indeed critical.

McNatton (1994) has emphasised just how complex the racing domain is and the importance of obtaining expert input when developing intelligent prediction systems in this domain. In addition, Tsang et al. (1998) have noted that data input is extremely laborious. Genuine expert input is very difficult to obtain[8] and, with very limited resources available for data entry, we found ourselves pretty-much forced to adopt Debenham's (op. cit) 'moderately expert' approach (noted earlier) and restrict our analysis to the following variables: *test, box, quote, tip, position* and *return. Form* is obviously critical and, from Figure 1, it can be seen that it is also quite complex. Our alternative was to use a particular expert's[9] tips as a surrogate for this variable.

RESEARCH DESIGN

This research was, essentially, exploratory in nature. The questions addressed were:

Q1: To what extent can modern artificial intelligence (AI) techniques be used to provide useful advice in complex and uncertain domains?

Q2: Can substantially-simplified DSS, constructed using these techniques, produce reasonable results (with the consequent benefit of reduced effort – in both DSS construction and operation)?

The AI technologies employed were rule-based expert systems, CBR, data mining and fuzzy logic. Additional functionality, based on traditional statistical analysis[10] was also employed. A number of other AI technologies, such as machine learning systems based on neural networks and genetic algorithms (Carbonell et al., 1983), were considered for use but rejected because of the substantial increases in complexity and effort this would have entailed (see *Q2* above). The greyhound racing domain chosen was introduced in the previous section and further detail is provided in the next and subsequent sections.

---

[6] These figures are for the previous 12 months and are readily available from newspaper form guides and the Web.

[7] Meaning that they may drop out the back early on, have a very easy run on the rails and, consequently, have plenty in reserve to enable them to come home strongly at the end.

[8] i.e. if a punter has real understanding of why one horse (or dog) is likely to win, why share that information? For the same reason, it makes no sense to purchase a 'winning guaranteed' betting system.

[9] A tipster called the 'Watchdog', whose tips for all meetings appear daily in the Melbourne Herald-Sun.

[10] Which also underpinned most of the AI techniques used to some extent.

The beauty of the domain used is, of course, that while the prediction process itself is highly-uncertain, outcomes are absolute: i.e. a dog either wins and/or places or it doesn't! Moreover, outcomes are (potentially) available within a few minutes of predictions being made. Finally, the key indicator of experimental outcomes is again absolute: i.e the amount of money won or lost.

Our experiment consisted of the following two stages:

1. A 'training database', constructed from 600 tips (involving 200 races) made by our chosen expert tipster was assembled between 4/2-8/3/2008. This database was analysed as detailed in the following section, resulting in the induction of four betting guidelines.

2. The four guidelines were then applied to a 'test database' of a further 1404 tips (468) races made by our expert, constructed between 11/3-31/5/2008. Profit and loss figures from this activity were calculated and analysed as discussed in the penultimate section.

It should be noted that, while our research appears to have produced some very-interesting results, these are preliminary at this point. In particular, substantial additional analysis will be required to validate tentative conclusions drawn from the initial application of our induced betting guidelines. On the other hand, we suspect that, in part at least, external validity may be strong: i.e. we see no reason why our results should not be applied to greyhound racing in other Australian states and, indeed, internationally.

THE DATABASE

The specification of the extensional database component is presented (in entity-relationship form) in Figure 2.

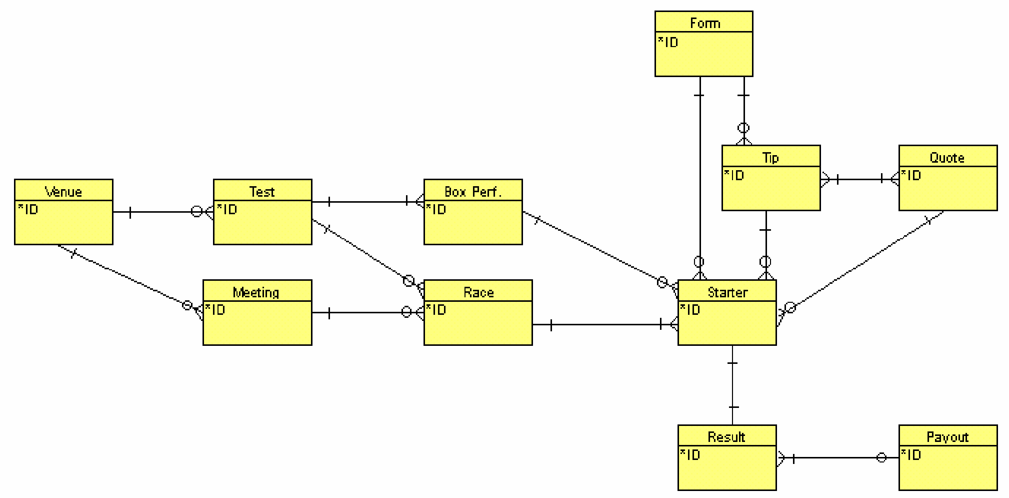

Figure 2: Database schema (extensional component).

A *venue* generally conducts many race *meetings* per year and each meeting consists of a number of *races* (10-12) over varying distances (generally no more than 2-3 distances, ranging from around 300m to 700m). A venue and distance together uniquely identify a *test* and associated with each of these is a set of *box performances.* All races, barring scratchings, have eight starters, each identified by a number corresponding to its starting box. A box performance, then, is the percentage of wins out of that box for the preceding 12 months.

Each starter has unique *form* (based largely on its recent performances in races) and this, relative to that of the other starters, will be the major influence on its *tip* (predicted finishing position). A *quote* (predicted payout for a win) is also associated with each starter and the quote and tip may mutually have an influence on each other. A *result* is a dog's finishing position in a race and there is a win and/or place *payout* associated with the first three finishers.

The database has been implemented in *Prolog* and this allows the specification of an intensional component: essentially virtual relations specified by rules which may also be employed as procedures to derive these relations (Kowalski, 1979). These are discussed further in the following section.

DATA ANALYSIS: TRAINING DATABASE

Rule-Based Approach

Using the *Flex*™ expert system shell (Westwood, 2007) we can represent our database as frames, specify domain knowledge in classic production rule form and, then, employ forward or backward chaining to infer consequences from facts. For example, we could specify a frame, *paying_result,* as a subtype of a *result* class and then generate an instance of this frame for every result where there has been a payout. An example is:

```
instance r0002 is a paying_result ;
      res_id is 2 and
      quote is 5.00 and
      pay_win is 12.10 and
      pay_place is 2.60 and
      return_value is average .
```

We could then declare that a return is considered *good_value* if the payment for a win is greater that the quoted odds. The relevant rule is:

```
rule good_value
      if R is some instance of paying_result
            and the quote of R is Quote
            and the pay_win of R is PayWin
            and PayWin > Quote
      then the return_value of R becomes good .
```

Once invoked, this rule will return all instances of *good_value* results and, in fact, with a little extra calculation code added, we may derive the fact that of the 122 training database races where there was a winner, only 36.9% represented good value. This is interesting as it may indicate that the quotes in the morning paper may be a little too high.

We may also represent rules in the 'raw' underlying *Prolog*. For example, we may be wish to find instances of non-winning streaks (sequences of races where the winner was not tipped). Sequences are most-naturally represented as lists and lists are the natural data structures of *Prolog* (Clocksin and Mellish, 1981: 41-48). Consequently, we may code our solution as the following recursive rules[11]:

```
R isa nonWinningStreak of 1 if
      R isa race and R1 isa race and
      R is the nextRaceAfter R1 and
      tip 1 won in race R1 and tip 1 lost in race R.

[R1|R] isa nonWinningStreak of N if
      R1 is the nextRaceAfter [R] and
      tip 1 lost in race R and
      [R] isa nonWinningStreak of N1 and N is N1 + 1.
```

We have used recursive rules of this type extensively within our system and found them extremely useful: for example, for analyzing and computing box advantages.

Case-Based Reasoning

CBR is largely based on the premise that, generally speaking, we don't think logically in *if-then-else* style but, rather, tend to try to match problem situations with previous cases (drawn from our memory) and then 'tweak' these cases to bring them into line with the current problem (Schank, 1975).

Assume that (as our 'target' case) we are interested in the place chances of a dog racing over 525m at the Meadows track (test 3). The dog has been quoted at $2.00, is tipped to finish first and has drawn box 1. This time, using our CBR toolkit (Shalfield, 2004), we may set up an input query on our training database as illustrated in Figure 3.

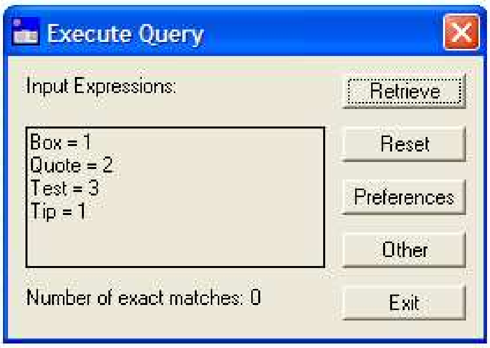

Figure 3: A CBR input query.

Note that (although, not shown in Figure 3) we have made it compulsory that that the tip must exactly equal 1 and that the test is 3. However, we have relaxed the constraints on the quote and box number:

[11] Represented, for expository reasons, as a 'quasi' *Prolog* procedure. This maps very easily to the underlying *Win-Prolog 4.700* syntax.

the former because we feel that there are unlikely to be many exact matches on *quote = $2.00* and the latter because we feel that 'close' to the inside is more important that an exact match on *box = 1.*

Output from the query (exported to *Excel™)* is presented in Table 2. The final column displays the CBR system's assessment of the degree of match between the input conditions and each retrieved case[12]. In 14/20 (70%) of these cases the dog placed, with an average return of $1.50. We may, therefore, be tempted to conclude that our target case represents a reasonable bet.

Table 3: Sample CBR query output.

Box	Quote	Test	Tip	PayPlace	Posn	Match %
1	2	3	1	1.4	2	99
1	1.7	3	1	1.04	2	99
2	2.5	3	1	1.3	1	98
1	2.7	3	1	0	0	98
2	2.8	3	1	1.5	1	97
2	2.8	3	1	1.4	3	97
1	2.8	3	1	1.2	1	97
1	2.8	3	1	0	0	97
1	3	3	1	0	0	96
3	2.8	3	1	1.8	1	95
3	2.6	3	1	1.8	3	94
1	3.3	3	1	1.4	1	92
1	3.3	3	1	0	0	92
3	3.3	3	1	1.5	2	90
3	3.3	3	1	1.4	3	90
3	3.3	3	1	0	0	90
3	3.5	3	1	2.2	3	88
3	3.5	3	1	0	0	88
1	3.6	3	1	1.5	1	87
1	3.6	3	1	1.5	1	87

The above is a fairly simple (but realistic) example of the application of CBR techniques. As a logical follow-up to this analysis, we may wish to restrict box or quote more tightly, relax restrictions on test or tip, weight one input variable (e.g. the quote) more highly than others etc. Although much of this can be done using CBR, however, this type of induction exercise is more suited to data mining. We address this in the following section.

Data Mining

Rule induction systems are probably the most common form of data mining application (Shalfield, 2004a). While CBR is fundamentally concerned with prediction, rule induction involves searching a database for 'interesting patterns' – which may then be expressed in the form: *antecedents* → *consequent.*

[12] Restricted to the top 20 matches only.

Results of one such search[13] are presented in Table 4. We are concerned here with the relationship between tips and boxes (antecedents) and results where there is a place payout (consequent).

Table 4: Data mining – tips, boxes and place results.

Measure	Tip = 1 Box = 1	Tip = 1 Box = 2	Tip = 1 Box = 1 or 2	Tip = 1 Box = 1,---,8
Base	201	201	201	201
Condnl	43	33	76	201
Hits	29	21	50	121
Accuracy	67.44	63.64	65.79	60.2
Coverage	21.39	16.42	37.81	100

Although the training database contains 603 *result* table entries, the *base* is actually 201[14]. Looking at the first column, there are 43 instances where a dog racing from box 1 was tipped to finish first and this is the *conditional count (condnl)*. Of these, 29 actually placed and this is the *hit count (hits). Accuracy* is *(hits/condnl).100* and, in this case, is 67.44%. *Coverage* is defined as *(condnl/base).100* and is 21.39% in our example. Accuracy is concerned with the extent to which one can rely on an induced rule and coverage is concerned with the extent to which a rule applies. There is a trade-off between accuracy and coverage and that is evident to some extent in Table 4.

In addition to starting boxes and tips, we have discussed a number of other factors that have in impact on whether a dog will win or place. We might summise that one of the most important of these is quoted odds and results of a data mining exploration focussed on the *quote* → *PayPlace* relationship[15] are presented in Table 5.

In Table 5, the trade-off between accuracy and coverage is demonstrated much more clearly. Where the quote is \$2.00 or less (column 1), accuracy is a very impressive 82.35% but this applies to only 8.46% of the base. If we then move the quote boundary out to \$3.00 (column 2), accuracy drops but coverage improves substantially. That is, the cumulative statistics indicate that, with the odds limit at \$3.00, we can now only expect a return in 65.43% of cases. On the other hand, we now have the option of betting in over 40% of races. This trend continues through columns 3 and 4.

As noted, in this preliminary research we have concentrated on place (rather than win) payouts. In addition to the impacts of tips, boxes and quotes on *PayPlace,* we also explored the effects of tests (broken down into venues and distances) and various combinations of all parameters. Space does not permit the presentation of all these results here but the more-critical relationships discovered were used

---

[13] Undertaken using *Win-Prolog's Data Mining* toolkit (Shalfield, 2004a).

[14] Because there are 603/3 (=201) races, with one tip each for places 1, 2 and 3 per race.

[15] Tips are also a factor in this exploration as we have restricted analysis to cases where the tip is 1.

to instantiate uncertainty factor parameters in our prediction engine. We now provide a brief introduction to how uncertainty may be dealt with in intelligent systems of this type.

Table 5: Data mining – quotes and place results.

Measure	Quote 0 - 2.00	2.01 - 3.00	3.01 - 4.00	4.01+
Base	201	201	201	201
Condnl	17	64	89	31
Hits	14	39	47	11
Accuracy	82.35	60.94	52.8	35.48
Coverage	8.46	31.84	44.28	15.42
Cumulative:				
Condnl	17	81	170	201
Hits	14	53	100	111
Accuracy	82.35	65.43	58.82	55.22
Coverage	8.46	40.3	84.6	100

Reasoning with Uncertainty

As noted earlier, modern expert systems are expected to be able to cope with uncertainty (to some extent anyway). Briefly, given a rule:

if Q_1 *and* Q_2 *then* P

we need to be able to deal with: i) imprecise data (Q_1 and Q_2); ii) rule uncertainty (how often do Q_1 and Q_2 imply P); and iii) imprecision in general. *Fuzzy logic* (Zadeh, 1968) is one popular approach to this problem.

Essentially, fuzzy logic provides a precise (or mathematically sound) means of dealing with real-world imprecision (Negoita, 1985). For example, we could specify that a dog's quoted odds are 'short' if they are $2.00 or less and establish rules based on that unambiguous fact. However, is there really much difference between a quote of $1.99 and $2.01? Fuzzy logic deals with this by assigning *degrees of membership* to fuzzy set elements. With our example, we could represent this graphically as illustrated in Figure 4. Thus, a quote of $2.90 is short to some extent but is much closer to an 'average' quote.

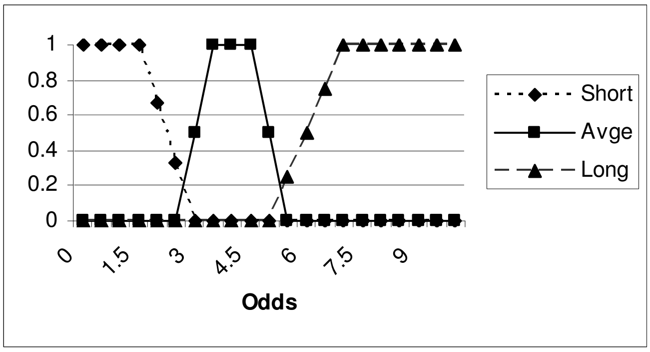

Figure 4: Flint™ specification of the fuzzy variable, odds_value.

Having also specified *box_adv* and *place_choice* as fuzzy variables, we may the use the *Flint*™ fuzzy logic toolkit (Shalfield, 2005) to declare the rules that combine instance of these as a 'fuzzy matrix'. The system then takes 'crisp' (actual) values for *odds_value* and *box_adv,* 'fuzzifies' these, calculates a fuzzy value for *place_chance,* 'de-fuzzifies' this and, finally, returns its calculated place probability (as a percentage). Data mining outputs were used to inform the critical fuzzy variable specification activity – particularly *box_adv*. It is intended that an extended version of the simple fuzzy logic system described here will be employed as an integral component of our prediction engine. At this early stage of our research, however, betting guidelines used for system validation have been expressed in crisp (rather than fuzzy) terms. We now turn our attention to validation activity.

PRELIMINARY BETTING GUIDELINES AND VALIDATION

Guidelines

The training database contains some 600 cases (tips/predictions) involving 200 races. Complete results of the analysis of the testing database are presented in Table 6.

Table 6: Test database – analysis results summary.

Quote	BoxAdv	Tip	Cases	Wins	AvWin	Places	AvPlace	Outlay	WTotal	PTotal
0-2.00	+	1	10	6	1.48	8	1.16	10.00	8.88	9.28
0-2.00	+	2-3	0	0	0.00	0	0.00	0.00	0.00	0.00
0-2.00	0	1	39	23	2.15	31	1.27	39.00	49.45	39.37
0-2.00	0	2-3	6	2	2.45	3	1.15	6.00	4.90	3.45
0-2.00	-	1	3	2	2.95	3	1.40	3.00	5.90	4.20
0-2.00	-	2-3	0	0	0.00	0	0.00	0.00	0.00	0.00
Group Totals			58	33		45		58.00	69.13	56.30
Pct				56.90		77.59				
2.01-3.00	+	1	47	17	2.48	35	1.37	47.00	42.16	47.95
2.01-3.00	+	2-3	27	13	2.93	17	1.45	27.00	38.09	24.65
2.01-3.00	0	1	162	59	2.81	109	1.47	162.00	165.79	160.23
2.01-3.00	0	2-3	82	21	3.58	46	1.54	82.00	75.18	70.84
2.01-3.00	-	1	13	6	2.67	9	1.41	13.00	16.02	12.69
2.01-3.00	-	2-3	4	0	0.00	2	1.65	4.00	0.00	3.30
Group Totals			335	116		218		335.00	337.24	319.66
Pct				34.63		65.07				
3.01-4.00	+	1	17	7	2.70	14	1.35	17.00	18.90	18.90
3.01-4.00	+	2-3	25	3	2.73	10	1.41	25.00	8.19	14.10
3.01-4.00	0	1	60	11	3.65	39	1.70	60.00	40.15	66.30
3.01-4.00	0	2-3	118	21	3.76	68	1.63	118.00	78.96	110.84
3.01-4.00	-	1	0	0	0.00	0	0.00	0.00	0.00	0.00
3.01-4.00	-	2-3	5	2	4.10	5	1.70	5.00	8.20	8.50
Group Totals			225	44		136		225.00	154.40	218.64
Pct				19.56		60.44				
4.00+	+	1	16	2	3.15	8	1.61	16.00	6.30	12.88
4.00+	+	2-3	90	13	6.35	41	1.95	90.00	82.55	79.95
4.00+	0	1	67	11	4.67	31	1.95	67.00	51.37	60.45
4.00+	0	2-3	480	58	6.92	210	2.03	480.00	401.36	426.30
4.00+	-	1	2	0	0.00	1	1.80	2.00	0.00	1.80
4.00+	-	2-3	50	4	9.65	18	2.21	50.00	38.60	39.78
Group Totals			705	88		309		705.00	580.18	621.16
Pct				12.48		43.83				
Totals			1323	281		708		1323	1140.95	1215.76

The analysis of the training database was covered in the previous section and this resulted in a number of conclusions, presented below as tentative guidelines:

G1: If the quoted odds are in the 0-2.00 range, bet for a win.

G2: If the quote is in the 0-3.00 range and tip = 1, bet for a win and place[16].

G3: If the quote is in the range 3.01-4.00 and tip =1, bet for a place only.

G4: If the quote is greater than 4.00, don't bet.

Essentially, this data represents the view (and performance) of our expert tipster and, in summary: i) of the 1323 tips, 281 were winners (21.24%) and 708 managed a place (53.51%); and ii) the overall ROI was 86.24% for a win and 91.90% for a place. Interestingly, our expert's performance compares very favourably with the panel of three tipsters employed by Chen et al. (1994) in testing their *ID3* system: specifically, their win bet ROI was only 65.0%.

Validation

Results of the application of guidelines *G1-G4* to our test database are presented in Table 7.

Table 7: Guideline application results summary.

Guideline	Cases	Wins	Win%	Places	Place%	Outlay	WTotal	PTotal
G1	58	33	56.90	*45*	*77.59*	58.00	69.13	*56.30*
G2	274	113	41.24	195	71.17	274.00	288.20	271.72
G3	77	*18*	*23.38*	53	68.83	77.00	*59.05*	85.20
G4	*705*	*88*	*12.48*	*309*	*43.83*	*705.00*	*580.18*	*621.16*
Actual Bets Only	409		98	248		683.00	357.33	356.92

Note: Italics indicate bets not place and where hypothetical data is provided for information only.

With respect to guideline *G1,* betting for a win only where the quote is in the $0-2.00 range, yields a return of $69.13 on a $58.00 outlay (ROI = 119.19%). This indicates that the strategy shows promise but, since only 4.38% of betting possibilities are within this quote range, it can only be applied rarely. It is also interesting that, even with very low average returns, place betting would result in an ROI of 97.07% (with a very high success rate of 77.59% - see Table 6). Obviously though, with so very few cases to work with, considerable care should be taken in basing any major investment decisions on this limited set of results.

On the other hand, guideline *G2* can be applied in 332 cases (25.09%). Here, win and place betting yields ROIs of 105.18% and 99.17% respectively, an overall return of 102.18%. Moreover, referring to

---

[16] Obviously, this overlaps with *G1* to some extent.

Table 6, it appears that win and place betting on tips 2 and 3 might also be appropriate in this range provided the dog has a box advantage. Again, however, further data is required.

With guideline *G3,* place betting on dogs tipped to win in the $3.01-4.00 range yields an ROI of 110.65%, suggesting the strategy has promise. As with *G1,* coverage here is low (77 out of 1,323 cases = 5.82%). Table 6 suggests though, that betting for a win in this range where there is a box advantage (and tip = 1) might also be worth further investigation.

Finally, our tentative suggestion that betting of any sort should be avoided where the quote is over $4.00 (guideline *G4)* appears sound, as the ROIs in this range are 82.30% for a win and 88.11% for a place. The downside is that 705 cases (53.29%) are covered by this guideline, a significant negative if a punter is betting principally for enjoyment and entertainment. It might also be worthwhile undertaking a finer-grained analysis of results within this range.

DISCUSSION AND CONCLUSION

Returning now to the research questions presented earlier, it would appear that our results do, indeed, suggest that modern AI techniques can be used to good advantage in complex decision making – at least, in the domain investigated. Specifically, application of our four induced betting guidelines yielded an overall return of $714.25 for a $683.00 outlay (ROI = 104.58%). This is substantially better than Chen et al. (1994) achieved with their *ID3* system and, in addition, is better than the performance of the expert tipster we employed as a surrogate for form (32.65% of winners tipped for an ROI of 90.86%).

Moreover, adopting this tactic (i.e. using a surrogate for *form* – and all the factors that impact on that variable) did enable us to greatly simplify the design, construction and use of our DSS. Notably, the intensional component of our knowledge base was reduced to a few simple rules and some accompanying *Prolog* code (used principally for calculation and list manipulation). Most of the complexity encountered during our study was in the rule induction process (conducted, primarily, as a data mining exercise) and most of the (hard-slog) work involved was in constructing the extensional component of the knowledge base (mostly routine data entry of key pre-race data and post-race results). Thus, the tentative answer to research question *Q2* is 'yes': i.e. a 'not so expert' system may well be able to provide good and useful advice in a complex and uncertain domain. In effect, of course, our approach has been to build upon the distillation of a specific expert's output without looking into the detail too much (a 'black box' approach in fact).

Much remains to be done however. Firstly, substantial extra data needs to be gathered and analysed in order to confirm the veracity of our guidelines and to explore the potential of additional observations made during testing (a couple of which were noted in the previous section). Secondly, there appears to be great scope for the application of alternative betting strategies (particularly those based on sequence betting) and it should be possible to do this without much additional data entry and maintenance costs.

Finally, as noted earlier, we suspect that our research results (and DSS) probably could be applied to greyhound racing in other Australian states and territories outside Victoria and, probably, internationally. The extent of the external validity of our results in other domains, however, is an open question – although, we expect that our system could well be adapted for application in stock market prediction and other financial forecasting applications.

REFERENCES

Abdel_Hamid, T. and Madnick, S.E. (1991). *Software Project Dynamics: An Integrated Approach,* Prentice Hall: Englewood Cliffs, NJ.

Asch, P., Malkiel, B.G. and Quandt, R.E. (1984). Market Efficiency in Racetrack Betting, *Journal of Business,* Vol.57, No.2, pp. 165-175.

Beemer, B. A. and Gregg, D.A. (2008). Advisory Systems to Support Decision Making, *Handbook on Decision Support Systems,* Springer: Berlin, pp. 511-528.

Berson, A., Smith, S. and Thearling, K. (1999). *Building Data Mining Applications for CRM,* McGraw Hill: New York.

Bonnistone, P.P. and Johnson, H.E. (1983). *Expert System for Diesel Electric Locomotive Repair: Knowledge-Based System Report,* General Electric Company, New York.

Buchanan, B.G. and Feigenbaum, E.A. (1978). DENDRAL and Meta-DENDRAL: Their Applications Dimension, *Artificial Intelligence,* Vol.11, pp. 5-24.

Carbonell, J.G., Michalski, R.S. and Mitchell, T.M. (1983). An Overview of Machine Learning, in (R.S. Michalski, J.G. Carbonell and T.M. Mitchell eds.), *Machine Learning, an Artificial Intelligence Approach,* Tioga Publishing: Palo Alto, CA, pp. 3-23.

Chen, H., Rinde, P.B., She, L., Sutjahjo, S., Sommer, C. and Neely, D. (1994). Expert prediction, Symbolic Learning, and Neural Networks: An Experiment on Greyhound Racing, *IEEE Expert,* Vol.9, No.6, pp. 21-27.

Clocksin, W.F. and Mellish, C.S. (1981). *Programming in Prolog,* Springer-Verlag: Berlin.

Cox, E. and O'Hagan, M. (1999). *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using and Maintaining Fuzzy Systems,* Morgan Kaufmann: San Francisco, CA.

Davenport, T.H. and Harris, J.G. (2007). *Competing on Analytics: The New Art of Winning,* Harvard Business School Press: Boston, MA.

Debenham, J. K. (1985). Knowledge Base Engineering, *Proceedings of the Eighth Australian Computer Science Conference,* Melbourne, 1985.

Debenham, J.K. and McGrath, G.M. (1983). Systems Development with Logic, *Proceedings of the Tenth Australian Computer Conference,* the Australian Computer Society, pp. 187-196.

Deliyanni, A. and Kowalski, R.A. (1979). Logic and Semantic Networks, *CACM,* Vol.22, No.3, pp. 184-192.

van Emden, M.H. (1978). Computation and Deductive Information Retrieval, in (E. Neuhold, ed.), *Formal Descriptions of Programming Concepts,* North Holland: New York, pp. 421-440.

Feigenbaum, E.A. and McCorduck, P. (1983). *The Fifth Generation: Artificial Intelligence and Japan's Challenge to the World,* Michael Joseph: London.

Hausch, D.B. and Ziemba, W.T. (1985). Costs, Extent of Inefficiencies, Entries and Multiple Wagers in a Racetrack Betting Model, *Management Science,* Vol.31, No.4, pp. 381-394.

Kim, K.J. (2004). Toward Global Optimization of Case-Based reasoning Systems for Financial Forecasting, *Applied Intelligence,* Vol.21, No.3, pp. 239-249.

Kowalski, R.A. (1979). *Logic for Problem Solving,* North Holland: New York.

Lee, R.S.T. (2004). iJade Stock Advisor: An Intelligent Agent Based Stock Prediction System Using Hybrid RBF Recurrent Network, *IEEE Transactions on Systems, Man and Cybernetics,* Vol.34, No.3, pp. 421-428.

Luger, G. (2005). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving,* Addison Wesley: Reading, MA.

McDermott, J. (1980). R1: A Rule-Based Configurer of Computer Systems, *Technical Report CMU-CS-80-119,* Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

McGrath, G.M. and Debenham, J.K. (1984). A Fifth Generation Pilot Project", *Proceedings of the Australian Computer Conference 1984,* the Australian Computer Society, pp. 389-403.

McNatton, S.W. (1994). *HOBBES: A Predicting Expert System for Thoroughbred Horse Racing,* Research Thesis, University of Kentucky.

de la Maza, M. (1989). A SEAGUL Visits the Racetrack, *Proceedings of the 3rd International Conference on Genetic Algorithms,* Morgan Kaufmann, pp. 208-212.

Minsky, M. (1975). A Framework for the Representation of Knowledge: The Psychology of Computer Vision, in (P. Winston ed.), McGraw Hill: New York, pp. 211-280.

Negoita, C.V. (1985). *Expert Systems and Fuzzy Systems,* Benjamin/Cummings: Menlo Park, CA.

Pereira, L.M. and Porto, A. (1982). Selective Backtracking, in (K.L. Clark and S.A. Tarnlund eds.), *Logic Programming,* Academic Press: London, pp. 107-114).

Quinlan, J.R. (1983). Inferno: A Cautious Approach to Reasoning with Uncertainty, *The Computer Journal,* Vol.26, No.3, pp. 255-269.

Rieger, C. (1975). Conceptual Menory, in (R. Schank ed.), *Conceptual Information Processing,* North Holland: Amsterdam.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction,* Springer: Berlin.

Segaran, T. (2007). *Programming Collective Intelligence: Building Smart Web 2.0 Applications,* O'Reilly Media: Sebastapol, CA.

Schank, R. ed. (1975). *Conceptual Information Processing,* North Holland: Amsterdam.

Shalfield, R. (2004). *Win-Prolog 4.700: Case-Based Reasoning (User Reference Manual),* LPA Ltd: London, UK.

Shalfield, R. (2004a). *Win-Prolog 4.700: Data Mining (User Reference Manual),* LPA Ltd: London, UK.

Shalfield, R.(2005). *Win-Prolog 4.700: Flint Reference Manual,* LPA Ltd: London, UK.

Shortcliffe, E.H. (1976). *Computer-Based Medical Consultation: MYCIN,* Elsevier: New York.

Tsang, E., Butler, J.M. and Li, J. (1998). EDDIE Beats the Bookies, *Journal of Software Practice and Experience,* Vol.28, No.10, pp. 1033-1043.

Tsang, E., Yung, P. and Li, J. (2004). EDDIE-Automation, a Decision Support Tool for Financial Forecasting, *Decision Support Systems,* Vol.37, No.4, pp. 559-565.

Westwood, D. (2007). *Win-Prolog 4.700: Flex Reference Manual,* LPA Ltd: London, UK.

Zadeh, L.A. (1968). Fuzzy Algorithms, *Information and Control,* Vol.12, pp. 94-102.