The Viability of Prudence Analysis

Richard Dazeley and Byeong-Ho Kang

School of Information Technology and Mathematical Sciences,
University of Ballarat, Ballarat, Victoria 7353, Australia.
School of Computing and Information Systems,
University of Tasmania, Hobart, Tasmania, 7001.
r.dazeley@ballarat.edu.au, bhkang@utas.edu.au

Abstract. *Prudence analysis* (PA) is a relatively new, practical and highly innovative approach to solving the problem of brittleness. PA is essentially an incremental validation approach, where each situation or case is presented to the KBS for inferencing and the result is subsequently validated. Therefore, instead of the system simply providing a conclusion, it also provides a warning when the validation fails. This allows the user to check the solution and correct any potential deficiencies found in the knowledge base. There have been a small number of potentially viable approaches to PA published that show a high degree of accuracy in identifying errors. However, none of these are perfect, very rarely a case is classified incorrectly and not identified by the PA system. The work in PA thus far, has focussed on reducing the frequency of these missed warnings, however there has been no studies on the affect of these on the final knowledge base's performance. This paper will investigate how these errors in a knowledge base affect its ability to correctly classify cases. The results in this study strongly indicate that the missed errors have a significantly smaller influence on the inferencing results than would be expected, which strongly support the viability of PA.

Keywords. Prudence analysis, knowledge acquisition, verification and validation, ripple-down rules, knowledge based systems

1 Introduction

Brittleness has been investigated from many possible sides. For instance, methodologies such as Knowledge Acquisition and Design Structuring (KADS) [1], were developed to help extract deeper forms of knowledge. Other systems such as Cyc [2] attempted to capture the majority of general knowledge to help soften the landing of KBSs when they left their core domain. A third area of investigation moved away from finding better methods of acquiring knowledge, and instead, developed a means of checking whether a KB was complete. This process is commonly referred to as Verification and Validation (V&V) [3].

V&V, however, is performed with known cases, where the inferenced results can be verified by an expert. Once the system goes online, not all cases can be checked by the expert and any errors by the system go mostly unnoticed. Prudence analysis (PA) is a relatively new, practical and highly innovative approach to solving the problem of brittleness. A PA enabled KBS has the ability to detect when an inferenced solution to a case may be wrong. While the few attempts at developing a PA system have had varying degrees of success, none are perfect. Occasionally a case is classified incorrectly and not identified by the PA system. The work in PA thus far has focussed on reducing the frequency of these missed warnings, however there has been no studies on the affect of these on the final knowledge base's performance.

This paper will investigate how these errors in a knowledge base affect its ability to correctly classify cases. It will do this by focussing on one of the more successful approaches and determining the classification ability of a knowledge base developed by a user that trusts the PA system. Firstly, a discussion of the various PA systems will be discussed followed by an explanation of the system being used in this study. Finally, the experiments performed will be discussed along with their implications on the viability of PA.

2 Prudence Analysis (PA)

V&V attempts to identify whether all the possible cases are covered by the KBS. Alternatively, in the subfield of anomaly detection a system is analysed holistically to find structural anomalies, such as redundancies, conflicts or dead ends [4]. PA, however, only uses actual cases as they are presented. Currently, PA has only been studied by a minority of researchers, all of whom have centred their studies on a single family of KBSs, referred to as Ripple-Down Rules (RDR) [5]. The primary reason for this is that RDR is an incremental KA and maintenance methodology with a flexible and maintainable structure.

The first work carried out relating to prudence checking was known as WISE by [6] and [7], which, after inferencing in standard RDR was complete, would search the RDR tree for repeated instances of the conclusion found. The paths to these conclusions were then compared to test the usefulness of storing the history of corrections. It was found that apart from adding to the explanation ability of the KBS it may be possible to also use such information for prudence checking [8].

WISE was later extended using reflective learning through what [7, 9, 10], termed prudence and credentials. The system developed was called Feature Recognition Prudence (FRP) which tested if further inferencing beyond the similarly matched conclusions was possible and resulted in different conclusions. Such situations indicated that there was a possibility of an error in the original conclusion found.

A second approach taken at the same time for prudence detection was Feature Exception Prudence (FEP), where, after a conclusion was generated, the system looked at the database and flagged features of the current case that have not been previously validated by the expert as permissible. If any flags are generated then the conclusion is identified as potentially invalid. These systems were used on the LabWizard ES showing the potential of finding all errors. However, it did suffer from

a large quantity of false positives [7, 9, 10]. A later study by [11] into prudence checking methods questions Edwards's results claiming they are highly reliant on the expert, order of cases and the nature of the dataset used.

While there are interesting features in Edwards's work, the fundamental problem is that there must already be knowledge within the KB for it to be able to predict missing knowledge, which does not truly solve the full prudence problem. [12] took a new approach of comparing cases with previously seen cases within context, and provided warnings if they differed in some unusual way. Basically, the method compared individual value-attribute pairs and warned if they exceeded what had been previously seen. This simple method achieved a reasonably high level of accuracy on some datasets with significantly less false positives. However, it concluded that the results were still not sufficiently accurate and still produced too many warnings to be applied in a real world application.

Recently two more studies have been conducted simultaneously. One is the basis of this study and will be discussed in detail in the next section while the other was a PhD thesis by [13]. This study continued [12] work by including two improvements. Firstly, the study included using attribute combinations to help reduce the amount of *false negatives*. The second improvement was to also develop a probabilistic profile of continuous attributes, which allowed a reduction in the number of *false positives*. This study's results indicate a reduction in false positives and more accurate warnings.

There is one significant difference between all of the above PA systems and the one used in this study. All the above systems based their warnings on the raw attributes of the case, either compared to the whole KB or in the context of the classification. Therefore, if a warning is missed for a particular case, it will always be missed for that case. This is because the case does not have the attributes required to cause the warning. The system used in this study produces warnings based on the structure of the knowledge base and the paths traversed during inferencing. Additionally, it learns which paths are used consistently. This allows the system to produce warnings for cases that earlier it may have missed.

3 Methodology

One problem with the above approaches is the reliance on attributes existence or absence in a case for warnings. This limit's the methods ability to be applied primarily in domains with a controlled number of only relevant attributes. Domains with large amounts of irrelevant attributes such as free text classification will tend to produce large amounts of false positives. The work in this paper has taken a significantly different approach. Instead of looking at attributes, it analysis the structure of the rule base and the paths followed by the inferencing process. Therefore, the method described in this paper is knowledge driven, rather than the previous approaches which were data driven.

The method developed is based on an algorithm built for classification and prediction, referred to as Rated MCRDR (RM). MCRDR (Multiple Classification Ripple-Down Rules) [14] is an extension of RDR developed to cater for multiple

conclusions per case. This section will first briefly overview MCRDR and RM. This will be followed by a discussion of how RM has been applied to prudence analysis.

3.1 Multiple Classification Ripple-Down Rules

Ripple-Down Rules is a maintenance centred methodology for a KBS based approach using the concept of fault patching [15] and was first proposed by [5]. It utilises a binary tree as a simple exception structure aimed at partially capturing the context that knowledge is obtained from an expert. The context is the sequence of rules that had evaluated to provide the given conclusion [5, 16-20]. Therefore, if the expert disagrees with a conclusion made by the system they can change it by adding a new rule. The new rule will only fire if the same path of rules is evaluated [18].

Ripple-Down Rules has been shown to be a highly effective tool for knowledge acquisition (KA) and knowledge maintenance (KM). However, it lacks the ability to handle tasks with multiple possible conclusions. Multiple Classification Ripple-Down Rules (MCRDR) aim was to redevelop the RDR methodology to provide a general approach to building and maintaining a Knowledge Base (KB) for multiple classification domains. The methodology developed by [14] is based on the proposed solution by [17, 18]. The primary shift was to switch from the binary tree to an *n-ary* tree representation. The main difference between the systems is that RDR has both an *exception* (true) branch and an *if-not* (false) branch, whereas MCRDR only has exception branches.

Knowledge is acquired by inserting new rules into the MCRDR tree when a misclassification has occurred. The new rule must allow for the incorrectly classified case, identified by the expert, to be distinguished from the existing stored cases that could reach the new rule [21]. This is accomplished by the user identifying key differences between the current case and each of the rules' cornerstone cases.

3.2 Rated MCRDR

The hybrid methodology developed in this paper, referred to as Rated MCRDR (RM), combines MCRDR with an artificial neural network (ANN). This function fitting algorithm learns patterns of fired rules found during the inferencing process. The algorithm is shown diagrammatically in Fig 1. Firstly, a case is presented to the MCRDR tree, which classifies the case. Then for each rule in the inference, an associated input neuron will fire. The network then produces a vector of output values, $\overline{v}$, for the case presented. The system, therefore, essentially provides two separate outputs; the case's classifications and an associated set of values.

Learning in RM is achieved in two ways. Firstly, the value for each corresponding value for $\overline{v}$ receives feedback from the environment concerning its accuracy. The network learns by either using the standard backpropagation approach using a sigmoid thresholding function, and the MCRDR component still acquires knowledge in the usual way. The only exception is when the expert adds a new rule to MCRDR. As the input space grows, new input nodes need to be added to the network in such a

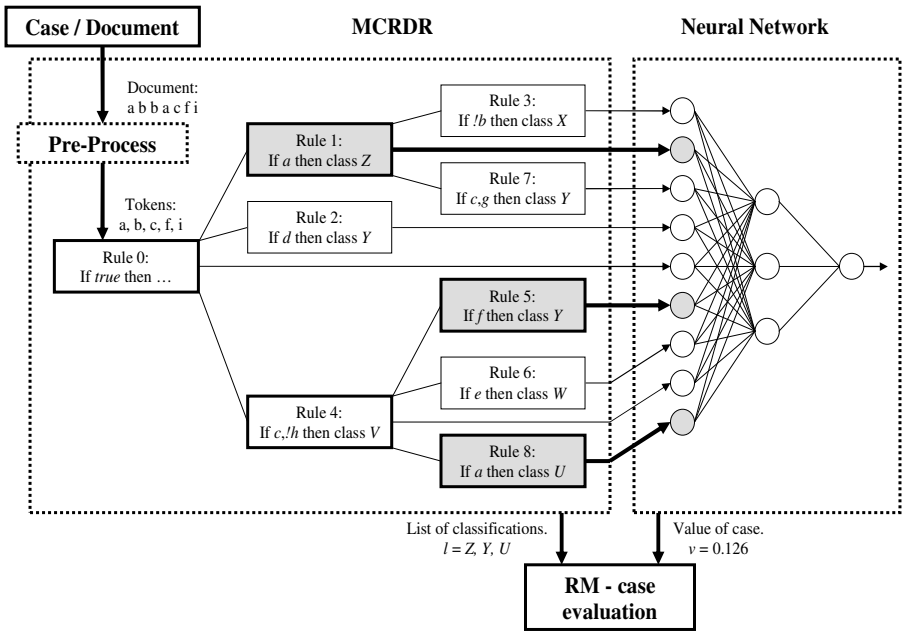

Fig. 1. RM illustrated diagrammatically.

way that does not damage already learned information. Therefore, the network structure needed to be altered by adding shortcut connections from any newly created input nodes directly to each output node and using these connections to carry a weight adjustment. When a new input node is added, additional hidden nodes are added. Fig 2, illustrates the process for adding new input and hidden nodes, it shows where new connections are added and what weights are given.

The *single-step-Δ-initialisation-rule,* Equation 1, directly calculates the required weight for the network to step to the correct solution immediately. This is accomplished by reversing the feedforward process back through the inverse of the symmetric sigmoid. It is possible for the expert to add multiple new rules for the one case. In these situations the calculated weight is divided by the number of new features, *m*. Finally, the equation is multiplied by the step-distance modifier, *Zeta (ζ)*. *Zeta (ζ)* should always be set in the range $0 \le \zeta \le 1$. It allows adjustments to how large a step should be taken for the new features.

$$w_{no} = \zeta \left[\left(\left(\log \left[\frac{f(net)_o + \delta_o + 0.5}{0.5 - (f(net)_o + \delta_o)} \right] \Big/ k \right) - \left[\left(\sum_{i=0}^{n-1} x_i w_{io} \right) + \left(\sum_{h=0}^{q} x_h w_{ho} \right) \right] \right) \Big/ m x_n \right] \quad (1)$$

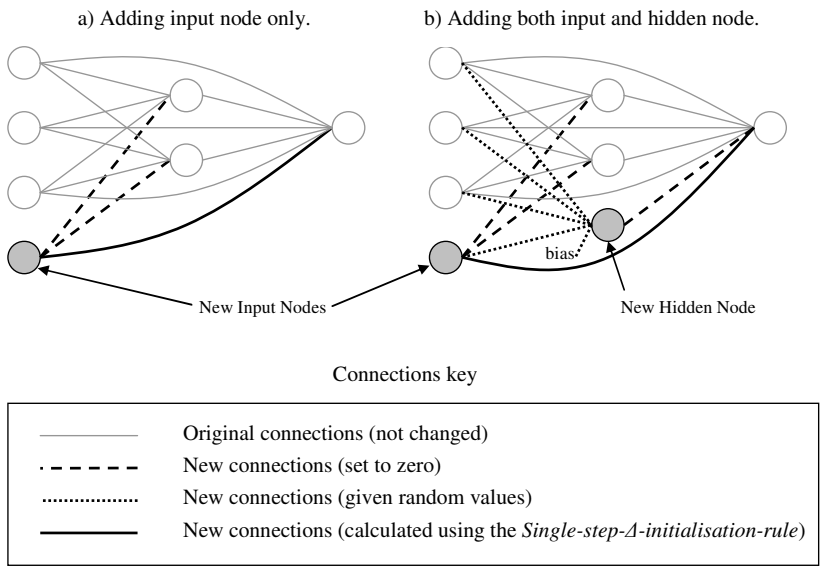

Fig. 2. Process used for adding new input and hidden nodes in RM. (a) shows how inputs are added by themselves. (b) shows how input and hidden nodes are added simultaneously.

3.3 RM Applied to Prudence Analysis

The basic idea behind applying RM to PA is to allow MCRDR to develop classifications in the general way, while the network passively watches rules being added to the MCRDR tree. As it watches it also attempts to identify the correct classifications. Through classification testing it was found that the classifications between the MCRDR component and the network often differed when MCRDR misclassified [22]. Therefore, the prudence system developed identifies these differences and warns the user that the classification by MCRDR could be wrong.

Training is a simple process of identifying the correct classification that the expert has agreed to when accepting a case. Obviously, however, this can only be done when a warning has actually been generated. When no warning is generated the system is unable to train because the system cannot be certain whether the expert would have wanted to alter the classification. When a warning is given and the expert confirms a classification the reward is a positive value at the output where it should have been classified as a particular case and a negative value otherwise.

Thresholding is performed on a per class basis. Basically, if the MCRDR and ANN classes were the same, then a warning was not generated. However, if the network's absolute rating for a particular class was below a certain threshold then it was interpreted as the network being unsure of its rating, and therefore, a warning would be generated. This second method of warning only occurred when the network had the same result as the MCRDR inference engine. This simple tool was found to be highly effective at improving prediction.

The thresholding value was found to only really be needed during the early phase of knowledge acquisition. Therefore, the thresholding value was also made dynamically adjustable. When a warning was warranted, the threshold was increased and when the warning was not needed, the threshold was reduced. Once again, it could not be adjusted when there was no warning generated.

4 Experimental Method

The method was previously tested and compared against [12], where it performed well [23]. The purpose of this study is to test whether the performance in this earlier study have a significant impact on the resulting knowledge based system. It is possible that even a relatively small amount of inaccuracy in a prudence based system could result in large problems in the knowledge base. For instance, when an expert does not correct a rule, because it was not warned about, any of the following could result:

1. The system may simply notice the next time a similar case arrives and warn the expert at this later stage. This option is not possible with the earlier forms of PA discussed in section 2.
2. The missed rule may never be created and the prudence system may not warn about future cases that may have also caused the creation of the rule.
3. The missed rule may have a compounding effect, where it causes rules that would have produced warnings later to now be missed because the knowledge base is damaged. This is a problem that potentially exists with the structurally based PA system in this paper but would not be expected in the earlier attribute based PA methods discussed in section 2.
4. The compounding effect could be exponential resulting in a KB that is woefully inadequate. This also would not be a problem in the earlier attribute based approaches to PA.

Without testing, it is unclear which of these results would occur. The greatest obstacle to the idea of prudence analysis using the structure of knowledge as a viable technique would be the compounding effect described in the last two points. The second issue will arise if the rules not created because of the missed warnings result in a KBS that is significantly damaged, which potentially would be the same for all PA methods. Therefore, the aim of this section is to determine the effect on the final KB when developed by a trusting expert. This has been tested by performing two separate experiments. The first result investigates the number of rules created to determine how well the KB progresses over time. The second test compares the trusted KB against the full KB when classifying unseen cases. Firstly, a description of the simulated experts and datasets used will be discussed.

4.1 Simulated Experts

One of the greatest difficulties in KA and KBSs research is how to evaluate the methodologies developed [24]. This is because any evaluation requires the use of people to actually build the system. Furthermore, the same experts would ideally be

used to compare two systems. Clearly, for this comparison to be meaningful they should also be built around the same domain. Therefore, the expert will have accrued some experience when building the first system and would provide better quality knowledge for the second system built. Even if the results of such a comparison could be gained in a meaningful way then finding an expert that is willing to provide their time to build two expert systems is next to impossible and would be prohibitively expensive.

The solution to this evaluation problem taken by the majority of RDR based research has been to build a simulated expert, from which knowledge can be acquired (Compton 2000). Generally, this has been accomplished by first building a KB using another KBS. The KA tool being tested can then use the simulated expert as its source of knowledge. This allows the method being tested to build a new KBS which should have the same level of competence as the original KBS. It is this approach that has been taken in this paper. This section will discuss the two simulated experts created for the tests performed in this paper.

4.1.1 C4.5 Simulated Expert

The first simulated expert created is similar to those used in other RDR research such as the one used by [12]. The only purpose of the simulated expert is to select which differences in a difference list are the primary ones. It uses its own KB to select the symbols that will make up the new KB. C4.5 [25] is used to generate the simulated expert's knowledge base. The resulting tree then classifies each case presented, just like our KB under development. If the KB being constructed, incorrectly classifies a case then the simulated expert's decision tree is used to find attributes within rules that led to the correct classification. Any attributes that appear in the different list and are in the path of rules between the root and the concluding leaf node, are used to select the attributes. A maximum of two attributes were selected from the top down. This is similar to the 'smartest' expert created by [26].

4.1.2 Multi-Class Simulated Expert

The fundamental problem with the above simulated expert is that it requires an induction system, such as C4.5, to generate a complete KB prior to its use. This is a problem because no such system is available that can create such a tree for a multiple classification domain. In C4.5, while there can be many conclusions, each case can only have a single associated class. However, the system being tested in this paper also targets multiple classification domains. Therefore, a second simulated expert was created specifically designed for handling a particular multiple classification based dataset.

This heuristic based simulated expert calculates classifications based on a case's attributes. The expert uses a table of randomly generated values, representing the level that each attribute, $a \in A$, contributes to each class, $c \in C$. Two rules were used in generating this stage of the expert's classification. Firstly, each attribute contributes to one class positively (1 to 3) and one class negatively (-1 to -2) and the remaining classes are given a value of zero. Secondly, each class has one positive and one negative attribute from every |C| number of attributes. An example of an expert's attribute table used is shown in Table 1.

	a	b	c	d	e	f	g	h	i	j	k	l
C1	0	0	-1	3	0	0	0	0	0	0	-1	3
C2	0	0	0	-2	2	0	0	-2	0	0	1	0
C3	0	-2	1	0	0	0	0	0	0	1	0	-1
C4	-1	3	0	0	0	0	1	0	-1	0	0	0
C5	0	0	0	0	-2	2	-2	0	2	0	0	0
C6	2	0	0	0	0	-2	0	1	0	-2	0	0

Table 1. Example of a randomly generated table used by the non-linear multi-class simulated expert. Attributes a - l are identified across the top, and the possible classes C1 – C6 down the left.

When a case is presented to the expert it is tested to see which class it belongs by adding all the associated values for each attribute in each class. The expert will then classify the case according to which classes provided a positive, > 0, total. The reason for setting up the expert in this way was to ensure that every case presented to the expert would be classified in at least one or more classes. When creating a new rule, the expert selects the attribute from the difference list that distinguishes the new case from the cornerstone case to the greatest degree.

4.3 Datasets

Below is a list describing each of the three dataset used from the University of California Irvine Data Repository [27]. Each has a brief description and details the number of attributes each case has.

- CHESS – Using the Chess end game of King+Rook (black) vs King+Pawn (white) on a7. This dataset has 36 attributes with a binary classification over 3196 cases.

- TIC-TAC-TOE (TTT) – This dataset uses the complete collection of possible terminating board configurations for Tic-Tac-Toe. This dataset has 9 attributes with a binary classification over 958 cases.

- GARVAN – This dataset uses a subsection of the full dataset that provided medical diagnoses for thyroid problems. This dataset has 29 attributes with 60 classification.

These three datasets were used as they are the same as those used by [12] study and thereby can be compared. A fourth dataset, the MULTI-CLASS dataset was also used to investigate how well the system performs in a multi-classification domain. The multi-class dataset had a variable number, between 3 and 9 of attributes per case. Therefore, it contained 3938 possible cases.

4.2 Rule Creation Comparison

This first test was performed to investigate how many rules are created in a system where the expert trusts the prudence analysis. In these experiments each dataset was run with ten different parameter settings. These were selected to give a range of performances. For instance in the first run parameters were set to produce the highest accuracy while in the 10th test it was set to the lowest accuracy for the system. In the experiment the rules were only corrected if there was a warning generated. Fig 3 shows the results gathered in four stacked area charts, one for each dataset.

The bottom area in each of the charts in Fig 3 show the percentage of rules created by the trusting expert against the total number a full expert would have created. In a system with full accuracy these would be the same giving an area of 100%. However, in systems with less accuracy this percentage would reduce. The second dark-grey area shows the percentage of rules we know were not warned about from the previous study [23]. If missed corrections are not fixed but all other rules are, as per the second point above, then the stacked percentage should be approximately 100%. This would be the theoretical result of the attribute based approaches from the earlier studies (section 2). If the missed rules are warned about later, as per the first point, then the percentage of the bottom area will increase resulting in the combined total for the two areas going over 100%. Thi is only possible in the structural based PA system in this paper and would indicate a significant advantage over the attribute based methods. If missed rules cause a compounding effect, where rules correctly warned about in the previous study are now missed because the knowledge base is incomplete, then the combined areas will drop below 100%. The attribute based approaches will not suffer from this and this would indicate a failing of the structural based approach.

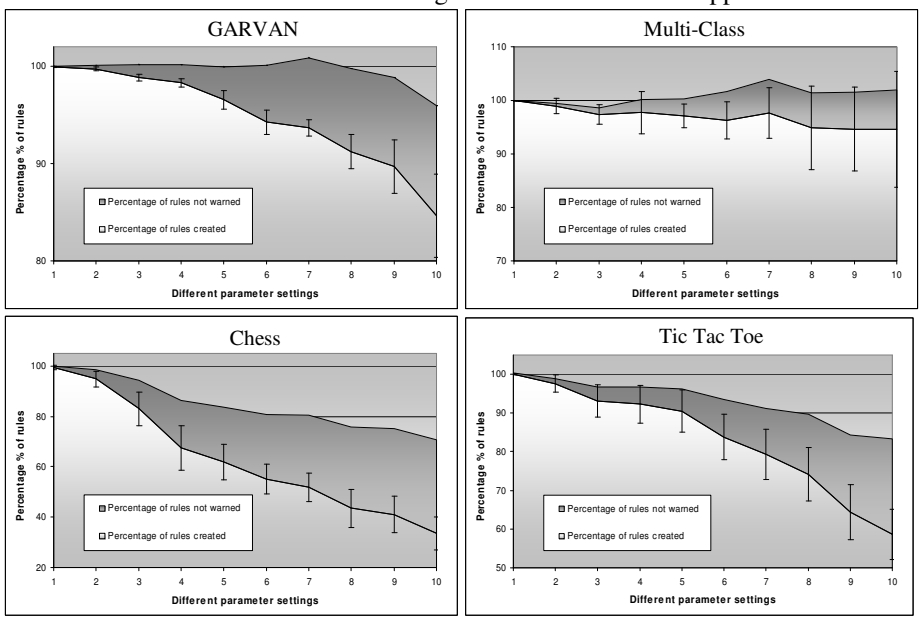

Fig. 3. Four stacked area charts showing the effect on a KB when rules are missed. Error bars show the 95% confidence range. They are not shown on the top area as they are too small.

Upon inspection, it can be seen that the results are mixed. Both GARVAN and the Multi dataset maintain a total area above 100% for the majority of parameter settings. This is an extremely promising result and indicates that the only rules missed are those not warned about and that some of these are even found later during the development cycle. This is a feature that would not be expected in the earlier attribute based systems as they contained no facility to find rules previously missed. However, the last two GARVAN results do start to degrade marginally, suggesting that this promising result is only possible when the accuracy is sufficiently high. The multi-class dataset does not show the same drop on average but the error bars do indicate that this performance was beginning to be affected by the order cases were presented.

Both the chess and TTT datasets' results, however, are not as impressive. It can be seen that they drop off relatively quickly. These show that unless near perfect accuracy is achieved then there is a severe compounding effect. Not only did these experiments miss the rules not warned about but the damaged KB caused additional results not to be warned. While this degrading performance does not appear to be exponential, it does appear immediate. Unlike the GARVAN and multi dataset which tolerated a degree of missed rules, the chess and TTT datasets dropped off the moment the degree of accuracy started to reduce. The issue with these datasets is that they are both constructed from rigid environments where the smallest change in the position of a piece can significantly alter the case. Therefore, missed cases are more likely to have an affect on the resulting knowledge base.

4.3 Classification Accuracy

The promising results in the last section show that the only rules not created are those not warned about and that even some of these are still found later on. This is important for the viability of structurally based prudence analysis, however, it does not show what effect even these few missed rules have on the classification ability of the resulting knowledge base. To judge the overall affect on the KB, a generalisation test was performed on each dataset with both the full and trusting experts. Fig 4 shows four charts showing nine classification tests. Each of the nine indicates how many 1/10th segments were used for training prior to being tested on the last 1/10th segment. The parameters chosen for each experiment all had the lower level of accuracy and therefore should result in the worst performance.

Interestingly, the trusted KBs performed nearly as well across each of the datasets. On the multi-class dataset the KB is almost identical while the GARVAN dataset only degraded by less than 1%. This is remarkable as the KB had lost approximately 7% of its rules because the expert did not receive a warning. Likewise, a similar performance can be seen on the chess and TTT datasets. For instance, the chess dataset only lost a little over 1% in its classification ability even though it had lost over 30% of its rule base and TTT had lost less than 15% of its classification even though it lost more than 40% of its rules.

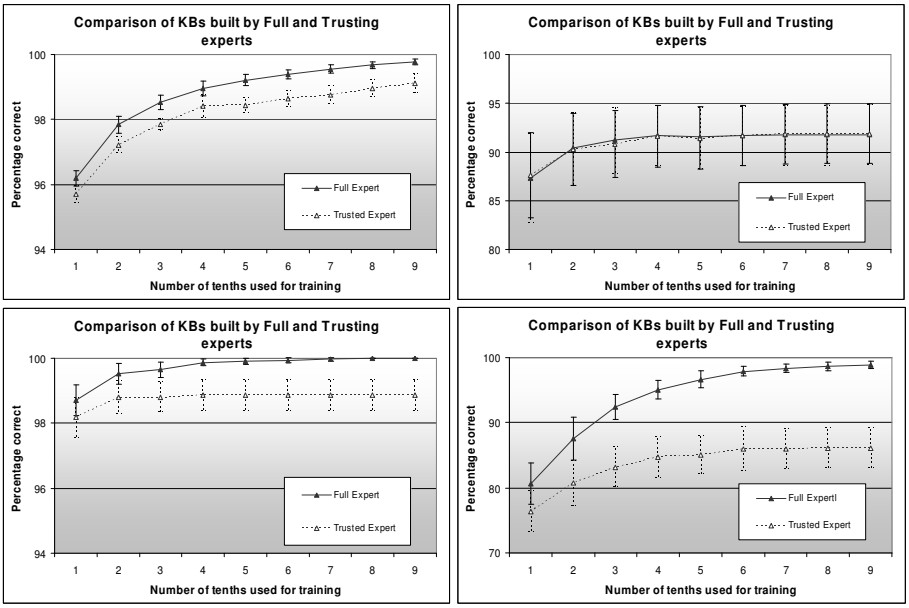

Fig. 4. Compares the full and trusting experts' knowledge bases after training. Each chart shows the KB's performance for the full system (where the expert checks every case) and the trusted system (where corrections are only made if a warning is generated). Error bars are also shown at a 95% confidence range. The full expert is shown with filled triangles and solid lines while the trusting expert has hollow triangles and dotted lines.

At first glance this seems remarkable that the KBs could still maintain such a high level of accuracy. This result though is most likely related to the general nature of KBSs. For instance, between 1984 and 1987 the GARVAN-ES1 KBS increased its number of rules by approximately 80% but only improved its level of accuracy from 96 to 99.7% accuracy [5, 28]. Therefore, like in the GARVAN-ES1 system, the rules RM fails to warn on appear to be the more specialised rules that only cater for the occasional rule. This can-not be proven in this study but the high degree of accuracy after losing many rules indicates that the rules remaining are more general in nature.

4.4 Non-Perfect Humans

The above results compared trusting the prudence system against an expert that checks every case. This is fine in a simulated environment. However, in a real world knowledge acquisition (KA) task, a human expert is rarely in a situation to fully check every case. It could be argued that, in fact the human expert is likely to miss the occasional misclassification in the full system. This is likely to be especially prevalent during the later stages of KA as they become more complacent towards the system's accuracy. It is entirely possible that the human expert could fail to notice more misclassifications than RM. For instance, it would not seem unlikely that a human

expert could easily miss 5-10% of errors in the KB. The results for RM, however, can achieve a much improved level of accuracy.

It could be further argued that a human expert is more likely to pay closer attention to a case when the prudence system produces a warning. Therefore, they are much less likely to miss a misclassification after a warning. Thus, it may be that in a real world system, the small reduction in performance of the trusted KB would be entirely dissipated or even reversed. Unfortunately, the performance of an expert in the two different environments is beyond the scope of this paper.

6 Conclusion

Prudence analysis represents a method for predicting when a case requires knowledge beyond the system's current KB. It is one way of attempting to resolve the issue of brittleness in current knowledge based systems. In theory, prudence analysis would be a very powerful tool when performing knowledge acquisition and maintenance. Previous work in this area, however, has yielded results that are not of sufficient accuracy, or that produce too many warnings, to make such a system viable.

Previously, no study had been undertaken to test the application of a prudence system and it was feared that the missed errors could compound through the KB development, resulting in a significantly flawed KB. Results, however, indicate that compounding of errors only occurs with very low levels of accuracy or in particularly rigid datasets where the smallest change in the case results in a different conclusion. In fact, some evidence was found to suggest that missed errors appear to be noticed and corrected in subsequent cases.

The most impressive result found in this paper was that, even though a number of rules may not be created because no warning was produced, a high level of classification accuracy could still be expected. For example, the KB built with the GARVAN dataset lost approximately 7% of its total rule base. However, this only caused a loss of less than 1% of the KB's classification ability.

It was further argued that in many application domains the resulting KB may even be better than one fully checked by an expert. This is due to the expert having less complacency when checking cases warned about by a prudence system than when every case must be verified. This advantage would be most prevalent when warnings are kept to a minimum. Therefore, the RM system presented in this paper represents a truly viable solution to prudence analysis.

Acknowledgements

The majority of this paper is based on research carried out while affiliated with the Smart Internet Technology Cooperative Research Centre (SITCRC) Bay 8, Suite 9/G12 Australian Technology Park Eveleigh NSW 1430 and the School of Computing, University of Tasmania, Locked Bag 100, Hobart, Tasmania.

References

1. Wielinga, B.J., Schreiber, T.,A., and Breuker, J.A., (1992) KADS: a modeling approach to knowledge engineering, *Knowledge Acquisition,* Vol. 4, No. 1, pp. 5-54.
2. Lenat, D.B., Guha, R.V., Pittman, K., (1990) Cyc: Toward Programs with Common Sense, *Communications of the ACM,* Vol. 33, No. 8, pp. 30-49.
3. Preece, A.D., (1995) Validation of knowledge-based systems: Current trends and issues, *The Knowledge Engineering Review,* Vol. 10, No. 1, pp. 69.
4. Kusiak, A., (2000) Computational Intelligence in Design and Manufacturing, Wiley-IEEE, pp. 535.
5. Compton, P., and Jansen, R., (1988) Knowledge in Context: a strategy for expert system maintenance, *Second Australian Joint Artificial Intelligence Conference (AI88),* Vol. 1, pp. 292-306.
6. Kang, B., (1996) Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules (PhD thesis),
7. Edwards, G., (1996) Reflective Expert Systems in Clinical Pathology,
8. Richards, D., (1998) The Reuse of Knowledge in Ripple Down Rule Knowledge Based Systems, pp. 336.
9. Edwards, G., Kang, B.H., Preston, P., (1995) Prudent expert systems with credentials: Managing the expertise of decision support systems, *International Journal of Biomedical Computing,* Vol. 40do have - RDR prudence and credentiuals is descussed WISE, FRP, FEP, pp. 125-132.
10. Edwards, G., Compton, P., Kang, B., (1995) New Paradigms for Expert Systems in Healthcare, *Proceedings of the Fourth Health Informatics of NSW Conference,* Primbee, pp. 67-72.
11. Naidoo, Y., (1998) Prudence in RDR, pp. 51.
12. Compton, P., and Preston, P., (1996) Knowledge based systems that have some idea of their limits, *10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW99),* SRDG publications, Canada,
13. Prayote, A., (2007) Knowledge Based Anomaly Detection,
14. Kang, B.H., (1996) Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules,
15. Menzies, T., and Debenham, J., (2000) Expert System Maintenance, *Encyclopaedia of Computer Science and Technology,* edited by A. Kent and J.G. Williams, Vol. 42; 42, Marcell Dekker Inc, pp. 35-54.
16. Beydoun, G., (2000) Incremental Acquisition of Search Control Heuristics (PhD thesis),
17. Compton, P., Edwards, G., Kang, B., (1991) Ripple Down Rules: Possibilities and Limitations, *6th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW91),* Vol. 1, SRDG publications, Canada, pp. 6.1-6.18.
18. Compton, P., Kang, B., Preston, P., (1993) Knowledge Acquisition Without Knowledge Analysis, *European Knowledge Acquisition Workshop (EKAW93),* Vol. 1, Springer, pp. 277-299.
19. Preston, P., Edwards, G., and Compton, P., (1993) A 1600 Rule Expert System Without Knowledge Engineers, *Moving Towards Expert Systems Globally in the 21st Century (Proceedings of the Second World Congress on Expert Systems 1993),* New York, pp. 220-228.
20. Preston, P., Edwards, G., and Compton, P., (1994) A 2000 Rule Expert System Without a Knowledge Engineer, *Proceedings of the 8th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff, Canada, pp. 17.1-17.10.
21. Preston, P., Compton, P., Edwards, G., (1996) An Implementation of Multiple Classification Ripple Down Rules, *Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop,*

SRDG Publications, Department of Computer Science, University of Calgary, Calgary, Canada UNSW, Banff, Canada,
22. Dazeley, R., (2007) To The Knowledge Frontier and Beyond: A Hybrid System for Incremental Contextual-Learning and Prudence Analysis, pp. 281.
23. Dazeley, R., and Kang, B.H., (2008) Detecting the Knowledge Boundary with Prudence Analysis, Vol. In Press, Springer,
24. Compton, P., (2000) Simulating Expertise, *Proceedings of the 6th Pacific Knowledge Acquisition Workshop,* Sydney, Australia, pp. 51-70.
25. Quinlan, J.R., (1993) C4.5: Programs for Machine Learning, Langley, P. ed., Morgan Kaufmann Publishers, San Mateo, California, pp. 302.
26. Compton, P., Preston, P., and Kang, B., (1995) The Use of Simulated Experts in Evaluating Knowledge Acquisition, *9th AAAI-sponsored Banff Knowledge Acquisition for Knowledge Base System Workshop (KAW95),* Vol. 1, SRDG publications, Canada, pp. 12.1-12.18.
27. Blake, C.L., and Merz, C.J., (1998) UCI Repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences,
28. Compton, P., Horn, R., Quinlan, R., (1989) Maintaining an expert system, *Applications of Expert Systems,* edited by J.R. Quinlan, Addison Wesley, London, pp. 366-385.