An Incremental Learning Method for Data Mining from Large Databases

by

Tristan Ronald Ling, BComp

A dissertation submitted to the

School of Computing

in partial fulfilment of the requirements for the degree of

Bachelor of Computing with Honours

University of Tasmania

November, 2006

I, Tristan Ronald Ling, declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution. To my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Tristan Ronald Ling

Abstract

Knowledge Discovery techniques seek to find new information about a domain through a combination of existing domain knowledge and data examples from the domain. These techniques can either be manually performed by an expert, or automated using software algorithms (Machine Learning). However some domains, such as the clinical field of Lung Function testing, contain volumes of data too vast and detailed for manual analysis to be effective, and existing knowledge too complex for Machine Learning algorithms to be able to adequately discover relevant knowledge. In many cases this data is also unclassified, with no previous analysis having been performed. A better approach for these domains might be to involve a human expert, taking advantage of their expertise to guide the process, and to use Machine Learning techniques to assist the expert in discovering new and meaningful relationships in the data. It is hypothesised that Knowledge Acquisition methods would provide a strong basis for such a Knowledge Discovery method, particularly methods which can provide incremental verification and validation of knowledge as it is obtained. This study examines how the MCRDR (Multiple Classification Ripple-Down Rules) Knowledge Acquisition process can be adapted to develop a new Knowledge Discovery method, Exposed MCRDR, and tests this method in the domain of Lung Function. Preliminary results suggest that the EMCRDR method can be successfully applied to discover new knowledge in a complex domain, and reveal many potential areas of study and development for the MCRDR method.

Acknowledgements

The drama of this year has been incredible, due in no small part to the production of this thesis. The research progressed well, the implementation became delayed, and Spring found me very nervous. With so much work to do in such a short space of time, on many occasions I couldn't see how I could possibly make it through. It is a testament to the people that were supporting me that I finally did.

First mention goes to Byeong Ho Kang, supervisor extraordinaire. Although he had even more work to do than I did for most of the year, he was always there to give me brilliant advice whenever I needed it. Thank you very much Byeong, I hope this work repays your faith in me in some way, and I hope I can continue to repay you by continuing this work next year.

To David Johns, co-supervisor, Lung Function expert, project client and general supporter: thank you so much for the friendly help you gave me in achieving this. From the outset you were incredibly positive and supportive about the whole idea, and I think it made all the difference. Many thanks also go to Justin Walls and Afshin Agahi, for their interest, advice, and support.

To the very fine people of StudioQ, Andrew, Nick, Chris, and Tony, I send a very big thank you. You not only provided a work environment that I looked forward to each day, where I could always learn, improve my skills, and be far too entertained, but you paid me to do it. You also deserve very special thanks for being incredibly understanding whenever I needed time away from work: I don't think this thesis could have been produced otherwise.

To Bob, more thanks go to you than I can express. You put up with a lot from me all year without complaint, provided more assistance and advice than I could ever have asked for or expected, and kept me focused and sane through everything. Put simply, I don't think I would've made it through the year without your help.

Clan FGI - a better bunch of blokes I just can't imagine. Along with all the Honours people from the last two years, you made those two years more fun than I thought

was possible. You certainly didn't make getting my thesis written any easier, but I guess I deserved that (“footy o’clock anyone?”). Now we're even.

Chris and Yoko, you also put up with a lot without complaint, so I have special thanks for you both. You kept me sane and always gave me something to look forward to. Sometimes it's nice to know that not everyone in the world knows or cares what a null reference exception is. Doumo arigatou gozaimashita. Watashitachi ha nihon ni ikimasu!

To my parents, you were the best taxi service money couldn't buy, but thank you for all the love and support. You know it means a lot to me, and that I would not be here without you.

And to Richard, my only older brother: I'm always thinking of you, and I know you can get through.

Table of Contents

1. Introduction

With the rise of computing technology for analysing and storing data, many fields are facing the difficulty of having vast stores of data about their processes which contain significant and useful information but which have not or can not be analysed to extract this information. Data Mining technologies were developed in order to extract meaningful information from these large stores of data, taking a place in the overall field of Knowledge Discovery: methods whereby new information is derived from a combination of previous knowledge and relevant data (Goebel & Gruenwald 1999). The subset of Knowledge Discovery which involved an expert, in order to make use of their domain expertise in determining how to discover new knowledge, has been labelled as Knowledge Acquisition (Gaines, B. R. 1993). More recently however the Knowledge Discovery field has been moving away from using human expertise and towards full automation of the process, due to efficiency constraints in the acquisition of this knowledge – the "knowledge acquisition bottleneck" (Buchanan & Shortliffe 1984). This is because one of the most common reasons for performing Knowledge Acquisition is to model how a domain works in order to facilitate the development of an expert system (Buchanan et al. 1983; Gaines, B. & Boose 1988; Liou 1990). Automated Machine Learning techniques have therefore become the main focus of research into Knowledge Acquisition (Grefenstette, Ramsey & Schultz 1990; Hong et al. 2000; Sester 2000). However certain domains contain data whose complexity is beyond the ability of Machine Learning algorithms to adequately and usefully explore (Abe & Yamaguchi 2005; Clerkin, Cunningham & Hayes 2001; Goldberg & Holland 1988).

One such domain is the clinical field of Lung Function testing. The lungs are a vital component of the human body: the continued function of a human requires that the lungs are operating effectively at all times (Ruppel 1994). However the processes by which lung function can be measured and dysfunctions identified are complex and not completely understood (Glady et al. 2003; Laszlo 1994; Swanney et al. 2004). There are vast amounts of data stored by respiratory laboratories around the world which have not been analysed to the greatest possible extent, but which can

potentially provide many beneficial insights into the field if such analysis were to be performed.

A further aspect of this data is that it has had no analysis performed on it in any way, and as such all cases are unclassified. There is also no standard for classifying cases in the domain. A reliable method of classifying the cases is required to assist in the further analysis, and would also be a useful tool in itself.

This study aims to show that a new method of involving and assisting an expert to perform Knowledge Discovery, derived from an MCRDR Knowledge Acquisition approach, can be applied to domains such as Lung Function where the data is unclassified, too complex for a human to extract information from without assistance, and the relationships too sophisticated and vast for machine learning techniques to find meaningful information from.

2. Background

2.1. Knowledge Discovery

Knowledge Discovery as a field covers techniques for finding and defining new information about a domain, using gathered domain data and/or existing domain knowledge. The field partially originated from studies into how to use large volumes of information to derive new and meaningful information (Frawley, Piatetsky-Shapiro & Matheus 1992), and partially from the study into more standardised methods for discovering new information in general (Gerber et al. 2004). It is a broad domain with many and varied applications, and many sub-domains of methods which attempt to perform the task in different ways.

That subset of Knowledge Discovery methods which do use compilations of domain data can be referred to as Knowledge Discovery in Databases (KDD) (Dazeley & Kang 2004; Frawley, Piatetsky-Shapiro & Matheus 1992). More generally, any methods for examining data for statistical trends and patterns, which may indicate a useful relationship that can be used in future work, are referred to as Data Mining methods (Leondes 2002; Witten & Frank 2000). These methods can either be automated, performed by a computer program with little or no understanding of the domain from which the data was gathered, or it may be a manual process performed by a human, either using expertise in the field to guide the search or simply analysing statistics and relationships without direction or guidance.

Goebel and Gruenwald (1999) show that effective Knowledge Discovery requires an understanding of the domain, and a validation and verification process afterwards, among other steps. This understanding of the domain, how relevant new information is to the domain, and how the information can affect the domain, requires expert involvement at some stage. In some domains, and with some datasets, the expert involvement can be minimal: simply providing enough background information about the domain to guide the analysis in an appropriate direction, or by pointing out features of the dataset which can be used as classifications, allowing comparative

analysis of sets of data. However, this is not an adequate approach for domains or datasets where the cases do not already have simple classifications, or where a direction of analysis can not be easily described.

In this situation there are two possible approaches: if the Knowledge Discovery process is a manual, expert-driven approach then the expert can personally determine which areas to examine at each step, and determine as progress is made how relevant the information is. This approach has the benefit of taking maximum advantage of existing domain knowledge to guide the search. However, this takes a significant amount of the expert's time; time which is usually quite valuable. The other approach is to use expertise to record existing domain knowledge in some form, and use this recorded knowledge combined with an automated analysis of the data to discover new relationships and meaningful information about the domain. The recorded knowledge may be as simple as giving each case a classification (i.e. defining known relationships between cases). There may also be no expert involvement before the analysis is carried out. In any of these situations, the end results produced must then be considered by the expert to determine whether the information is meaningful, new, or otherwise useful; the knowledge must then also be distributed to programs or people who can make use of it in working in the field (even if the distribution is only to the witnessing expert themselves).

To facilitate the entirely expert-driven approach, many user-driven statistical data analysis tools have been developed, for a large number of domains (Witten & Frank 2000). These are generally simple programs that assist the user by providing statistics about data upon request. However, while tools such as this might help an expert to categorise and discover connections within the data, they lack any element of knowledge about the data they are categorizing. That is, these tools cannot assist the expert in knowing which data and correlations are more meaningful, or even in recording or testing new hypotheses, leaving the expert with no direction as to where to start the examination and no support as to which statistical relationships are particularly relevant. This problem contributes to an overall failing of these methods: that the process is too time-consuming to be capable of discovering significant information.

Methods of automated domain modeling are referred to as Machine Learning techniques, which are discussed in Section 2.4 Machine Learning. As these techniques generally require some form of previously modeled expertise (Gaines, B. & Boose 1988; Witten & Frank 2000), they also require a method of extracting that expertise: this expertise extraction, whether for the purposes of use with a Machine Learning algorithm or for the explicit modeling of domain knowledge, is known as Knowledge Acquisition.

2.2. Knowledge Acquisition

Knowledge Acquisition is a closely related field of study to Knowledge Discovery, with a similar goal and more specialised approaches. Knowledge Acquisition has been defined as the process of "extracting, structuring, and organising knowledge from human experts so that the problem-solving expertise can be captured and transformed into a computer-readable form" (Liou 1990). Another way to express the idea is that Knowledge Acquisition is the process of modelling human expertise within a domain. The knowledge attained in this way is then typically used as the basis for an expert system which can perform or support some of the tasks of such an expert (Buchanan et al. 1983; Gaines, B. & Boose 1988), but can also be used as the basis for any technique which requires domain expertise in order to function.

2.2.1. Knowledge Acquisition Methods

2.2.1.1. Classification Rules

Classification rules are possibly the simplest of the data modeling and Knowledge Acquisition techniques. While classification rules can either be an automated Machine Learning process, or they can be built using an expert's domain expertise. In this case, the expert examines the dataset and creates rules which classify cases based on the values of the set attributes. For example, all cases with attribute A above 30 and where attribute B is negative should have conclusion 1 (if A>30 AND B<0 then 1). The process can be partially automated by having an expert define which attributes should be considered for rule conditions, and selecting another attribute to be used as the conclusion: pure statistical analysis can then find what

values are required for the condition attributes for each value of the conclusion attribute (according to the specific dataset, which can hopefully be generalized to the domain). Rules can then be automatically generated based on statistical similarities between attributes, and between cases that have the same, or similar, attributes within the dataset (Roberto J. Bayardo & Agrawal 1999). However, if the goal is to discover entirely new knowledge the expert may not be able to narrow the list of attributes to use sufficiently, and this will result in a list of rules far too extensive for the expert to be able to sort through and determine which rules are valid, which do not describe interesting information, and which are worth considering for further examination (Bachant & McDermott 1984; Barker et al. 1989). Completely autonomous rule generation, based entirely on statistical relationships, suffers from the same problem to a greater extent. This method is also prone to generating very simple rules which have no relevance and very complex rules based on dataset-specific, coincidental relationships (Towell, G. & Shavlik 1994).

The major advantage of classification rule systems is that the structure of the knowledge learned is readable by the expert – if the expert wants to know why a classification was made, they can simply examine the clauses of the rule that fired (Clancey, William J. 1984). The expert can also easily view the compiled knowledge and see what conclusions are being made based on what information, hence providing a simple means to review effectiveness and progress. The expert can also easily build on this by modifying parts of existing rules, combining rules, or specifying default conditions for new rule generation.

It has also been noted that Classification systems provide an excellent framework for Knowledge Acquisition, in that it is easy for the expert to provide a classification for a case, and a rule defining why this classification should be made (Clancey, William J. 1984; Compton & Jansen 1989).

2.2.1.2. Decision Trees

Using the decision tree method, a logical tree is formed consisting of *nodes* and *branches*. An attribute is associated with each node, and for each possible value (or range of values) for that attribute a branch is created leading to a lower node. The lowest nodes have no outward branches, and contain a classification (Witten & Frank

2000). In this manner, a case can be presented to the tree, and by following the branches appropriate to the values for the case, a classification is found. Hence, "knowledge" is stored in a relatively simple to follow format, and one which can easily be transformed into a graphical representation (Quinlan, JR 1986). These features make it easy for the expert to understand how the system comes to a classification, and easy for the expert to have input into the way that knowledge is structured at each step. The tree can also be derived automatically, but only if the cases in the dataset already have a classification attached to them (Witten & Frank 2000).

The drawbacks of the method are that for complex classifications involving many attributes, the tree can get very large and convoluted. Another disadvantage is that similar cases can follow very different paths through the tree to get to roughly the same conclusion. Also, some classifications may be based on only a few attributes, or many, making it difficult for the expert to quantify exactly which attributes lead to which conclusions and what relationships exist between different conclusions (Quinlan, J. R. 1987; Witten & Frank 2000).

2.2.1.3. Case Based Reasoning

Another common Knowledge Acquisition technique is Case Based Reasoning (CBR), in which knowledge is represented by a set of stored cases, and a set of defined classifications, all of which is determined to sufficiently represent domain knowledge (Aamodt & Plaza 1994). This is a relatively simple method of storing knowledge about the domain, and one which is generally very easy to display in an understandable format. It has been used as a KA technique, but has been sufficiently effective only when combined with other techniques (Féret & Glasgow 1997; Golding & Rosenbloom 1996; Manago et al. 1993; Yamaguti & Kurematsu 1993). This is required to overcome the inherent problems within CBR: specifically, the problems involved in determining what constitutes a similarity, and the difficulty in directly influencing the results that the method provides (Ihrig & Kambhampati 1995). The Knowledge Acquisition is performed by the expert examining cases individually, and inputting them, case by case, into the system. The CBR method compares the current case with all the previously stored cases in the knowledge base, and produces a list of cases that it considers (based on threshold values) to be similar

to the new case (Aamodt & Plaza 1994). The method would then use the conclusion(s) of those similar cases as the conclusions for the new case. The expert examines this, comparing it with their own opinion of what the conclusion should be for that case (Kolodner 1991). If the expert believes there is an error in, or something missing from the logic the system is using the expert corrects or adds this knowledge as appropriate, in the form of adding the current case to the set of stored cases, and attributes or changing which conclusions apply to it. This can also be performed as a Machine Learning method, even on datasets without classifications attached to the cases (Watson & Marir 1994).

The CBR system can provide reasoning for any classification it makes, by presenting the past cases that were used to generate that classification, and the attributes that were similar enough for the system to associate the two cases. The resultant knowledge base provides the expert with information on which attributes are commonly linked, and which attributes can lead to common classifications. If the expert has a particular interest in some attributes, those attributes can be weighted higher than others when comparing a new case to the stored cases, or if the expert believes certain cases to be more typical than others in the domain, they can weight those individual cases more heavily (Kurniawati, Jin & Shepherd 1998; Watson & Marir 1994). This allows the expert greater flexibility in analysing the data to try and discover new information. CBR algorithms can generally be successfully applied to any learning situation (Aha 1991), which is an important feature when attempting to discover new information. In some domains, the functioning of CBR systems have been found to be more representative of the manner in which experts perform their tasks, and hence Knowledge Acquisition is a much easier process for the expert (Kowalski 1991).

A common drawback of CBR is that knowledge based entirely on previously seen examples may only represent a small subset of the dataset, rather than representing the entire domain (Chi & Kiang 1991). While this can be said of any technique using a dataset, it is particularly apparent in CBR due to the method being based entirely on the cases themselves.

2.2.1.4. Ripple Down Rules

As well as these standard methods, there are also others which combine techniques already discussed, or make specific modifications to those techniques. For example, Chi and Kiang discuss a combination of CBR and classification rules (Chi & Kiang 1991). A similar method that is of particular interest to this study is Ripple-Down Rules (RDR). RDR also combine elements of CBR and classification rules, but focus on building the knowledge base through expert use the system, without any knowledge engineer assistance (Bindoff 2005; Compton, Cao & Kerr 2004; Kang, Compton & Preston 1995). This method does involve the expert at every stage however, and excepting some attempts at automation (Gaines, B. & Compton 1992; Kang, Compton & Preston 1998) this method is a Knowledge Acquisition method, and not a Machine Learning method.

The knowledge base of RDR is built in a tree format, similar to a decision tree, but one in which every node contains a rule and a classification. The rule is a rule structured in the same way as a classification rule, i.e. conditions pertaining to the values of attributes in the dataset. When a case is presented to the system for classification, it is compared to the rules for nodes at the top level of the tree. If a rule is satisfied for the current case, then the child nodes of that node (defined as exceptions to the parent rule) are considered and their rules tested against the current case. Note that the rule for any node therefore implicitly includes the rules of all ancestor nodes. This continues until no further child nodes' rules are satisfied, or no further child nodes exist. The case is classified according to the classification associated with the last node whose rule was satisfied (Compton & Jansen 1989).

The key is in the addition of new nodes to the knowledge base. When the system attempts to classify a case but produces an incorrect classification, the expert is asked why they consider the classification to be incorrect. The expert chooses from the list of differing attributes those which are responsible for the different classification (Compton et al. 1993; Kang & Compton 1992; Richards & Compton 1997). This information can be added as the rule for the new node because, as noted before, child nodes are always considered entirely within the context of the parent node (i.e. if a node is being considered, then all ancestor rules for that node must have been satisfied). In this way, RDR can be regarded as a set of rules with exceptions (Catlett

1992; Compton et al. 1992; Kang, Compton & Preston 1995). The knowledge acquisition structure is summarized in **Error! Reference source not found.** below.

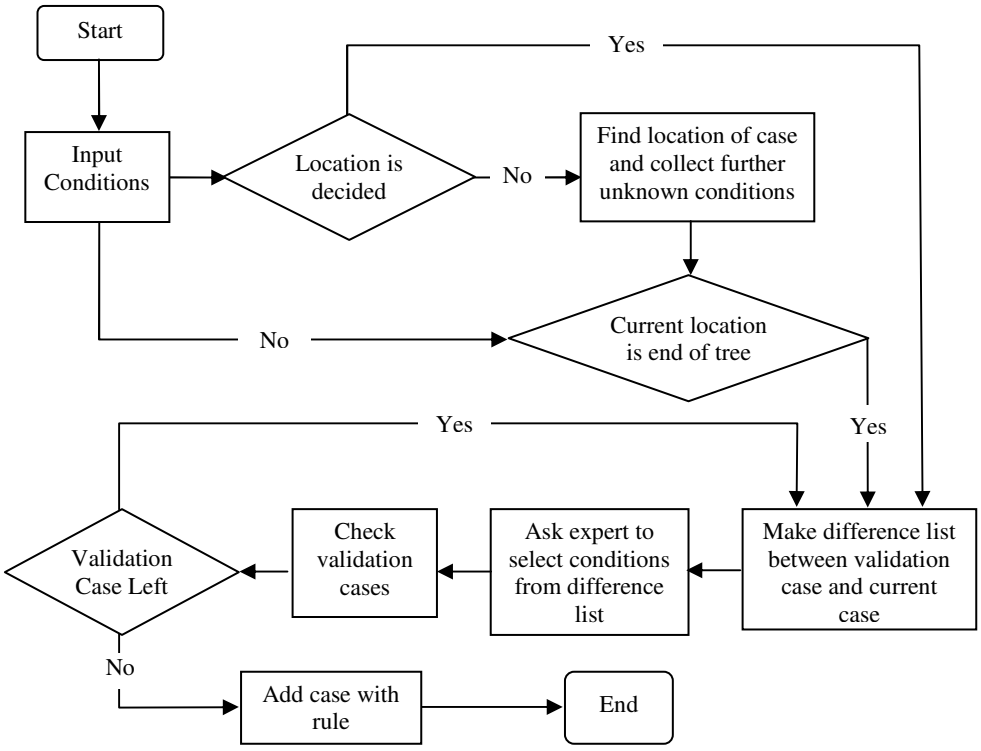

Figure 1: MCRDR Knowledge Acquisition Process (Kang et al. 1997)

This KA method was derived by studying how experts perform classifications in their normal work: an expert will produce a classification, and when asked how they came to that classification, they will justify why their classification is correct (Compton & Jansen 1989). Further, the values the expert uses in these justifications are dependant on the context within which the question was asked (Clancey, W J 1993; Compton et al. 1993). These findings cast doubt on the abilities of traditional Knowledge Acquisition techniques to accurately and fully extract expert knowledge, as they are largely centred around trying to determine the content of the knowledge base in a holistic fashion, meaning that they do not provide the expert with the full context within which they are working in any given instance, and so frequently miss many of the finer details of the domain (Compton et al. 1993). Also, almost all KA techniques are exclusively dependent on a knowledge engineer defining the knowledge base (Paris & Gil 1993), which adds a level of abstraction which, in the worst case, can potentially invalidate the entire purpose of involving an expert at all as the knowledge engineer's interpretation of the expert's descriptions will be the

"knowledge" that is entered into the knowledge base. However, it has been shown that building a knowledge base incrementally, with each step provided fully in context is a viable approach (Compton & Edwards 1994), and that it is in fact beneficial in that it allows the knowledge base to be built as it is in use. RDR Knowledge Acquisition works with this apparent predilection of human nature to define a knowledge base that is easy to add to and easy to understand. Although the knowledge base may appear convoluted when looked at as a whole, it is fast for the system to use and produce a result – only one path through the tree needs to be taken at any time. Similarly, if the knowledge base is queried about why it has made a classification, it can respond in easily understandable terms – providing a justification for the result. Richards and Busch (2003) noted that this means of knowledge acquisition also automatically (if gradually) discovers tacit knowledge about the domain – knowledge that the expert uses regularly but which the expert finds difficult to, or cannot, express explicitly.

Although the knowledge base builds easily and automatically as the system is in use, a major disadvantage is the considerable time that it can take to build a knowledge base that covers the domain (Richards & Compton 1997). Another problem is that classifications may have to be entered multiple times, if the same classification occurs as exceptions to multiple contexts (Kang, Compton & Preston 1995). One of the most significant drawbacks is that only one classification is allowed for any one case, even though the domain may describe many classifications for a single case. Even if the system contains rules to classify the case in multiple ways, it will simply use the first classification that it reaches (Compton et al. 1993). Therefore, if a case is to have multiple classifications it would require a rule with a compound classification, that is, a single classification that describes multiple problems. This introduces a potentially exponential increase in knowledge acquisition time for domains which require multiple classifications (Kang, Compton & Preston 1995).

The most well known and possibly most successful RDR system is PEIRS (Pathology Expert Interpret Report System). PEIRS is an expert system which uses chemical pathology reports to produce clinical diagnoses (Compton et al. 1992; Edwards et al. 1993; Srinivasan et al. 1992). When it first entered into use PEIRS contained approximately 200 rules. However, through routine use the knowledge

base has been constantly updated: as of 1998, there were 1800 rules (Kang, Compton & Preston 1998).

2.2.1.5. MCRDR

To overcome the flaws associated with RDR, the technique was revised to Multiple Classification Ripple-Down Rules (MCRDR) (Kang & Compton 1992). The most significant functional change is that the technique can produce multiple classifications for a single case without the use of compound classifications. This is achieved by evaluating every node at the top level and following through all valid paths, rather than simply following the path of the first rule that matches (Kang, Compton & Preston 1995).

As a result of this, Knowledge Acquisition in MCRDR must also be handled slightly differently. If the expert determines that a case has been classified incorrectly, or if the system has not produced a classification for the case, then the system requests the correct classification from the expert. Once this has been recorded, the system must determine where within the knowledge base to place the new case, and develop an appropriate exception rule (Kang, Compton & Preston 1995). Extracting the correct classification(s) from the expert is identical to the RDR method, except potentially on a larger scale. If the system produces classifications A, B and C, and the expert determines that the correct classifications should be A and D, and not B or C, then the system will ask for a valid justification of why B is incorrect, why C is incorrect, and why D should have been found. Forming the rules is relatively trivial, as the system can provide the expert with a series of attributes which can be used as differentiating factors, and the expert can make relevant choices. The difficult task then becomes placement of the new rule into the knowledge base. This involves determining whether a new rule is in fact an exception to a current rule, whether the current rule is simply incorrect in some circumstances and must be stopped, or whether the new rule represents entirely new knowledge that requires no previous context (see Error! Reference source not found. for this process).

Table 1 below summarises these three situations, and the appropriate action to take. Of note are the terminating rules, added when a classification needs to be stopped:

the rule defines the circumstances which need to be met for the rule to be considered incorrect (Kang, Compton & Preston 1995).

Wrong Classifications	To correct the Knowledge Base
Wrong classification to be stopped	Add a rule (terminating rule) at the end of the path to prevent the classification
Wrong classification replaced by new classification	Add a rule at the end of the path to give the new classification
A new independent classification	Add a rule at a higher level (to the root) to give the new classification

Table 1: The three situations in which new rules can be added to a knowledge base (Kang, B., Compton & Preston 1994)

Besides potentially increasing the amount of work required by the expert in adding new knowledge to the knowledge base, none of these changes particularly add any difficulty of use to the system. However, the ability to make multiple classifications is crucial to most domains, particularly in the medical and diagnostic domains where a patient is not restricted to having a single abnormality or illness, and drawing broad conclusions based on a single classification is extremely unlikely to produce accurate or beneficial results.

2.3. Knowledge Engineering

A closely related field to Knowledge Acquisition is Knowledge Engineering. Where Knowledge Acquisition is the process of discovering and recording expertise, Knowledge Engineering is the process of creating a framework to put that recorded expertise into use (Feigenbaum 1977) (Feigenbaum in fact describes Knowledge Acquisition as a subset of the overall field of Knowledge Engineering). This is a vital component of Knowledge Acquisition: without a structure suitable to the purpose of the Knowledge Acquisition, the expertise will not be as accessible and useful as it could be. Liou (1990) also describes three primary factors to consider when performing Knowledge Acquisition: involving the correct people, primarily domain

experts and knowledge engineers; using proper techniques to elicit the knowledge; and a structured and systematic approach to performing the Knowledge Acquisition.

2.4. Machine Learning

Although these Knowledge Acquisition methods showed success in some applications, research and development in the expert system area discovered that the most significant problem faced, negatively impacting on both the effectiveness and cost of creating an expert system, was the Knowledge Acquisition phase. The so-called "Knowledge Acquisition Bottleneck" (Buchanan & Shortliffe 1984) caused a change of attitude in this area of Knowledge Discovery, shifting the focus from trying to model human expertise towards the more automated processes of Machine Learning (Grefenstette, Ramsey & Schultz 1990; Hong et al. 2000; Sester 2000), which is, using statistical analysis to derive knowledge about how the domain functions (Witten & Frank 2000). This has the major benefit of being able to create an expert system or to derive domain knowledge by analysing collected data, with limited expertise required: removing the requirement of having an expert in the domain take considerable time to develop the knowledge. This is of particular benefit in many domains where an expert's time is quite valuable. Machine Learning methods also allow the possibility of discovering the knowledge in a different manner to the way in which the expert would describe it – this may be an advantage or a disadvantage, depending on the domain and the ability of the experts to communicate domain knowledge: for example, the method may discover relationships that would otherwise go unexplored because the current expertise in the field does not suggest any such relationship could exist; or it may be a disadvantage, because relationships may be discovered which are present in the dataset but which are not present in the wider domain; or it may be disadvantageous because the method of discovering the relationships may be less efficient, effective or comprehensible than those used by an expert.

2.4.1. Machine Learning Methods

2.4.1.1. C4.5

C4.5 is a well known Machine Learning algorithm, which automates decision tree generation (Quinlan, J 1993). Various extensions have been made to the method to improve the efficiency, effectiveness, and generality to the domain of the trees produced (Quinlan, JR 1996). However this method has a number of disadvantages: the most significant drawback of C4.5 and similar methods is that they can not incrementally learn knowledge. The tree learned is derived from the set of data presented, and can not be easily modified. Also, the method can not incorporate previous domain knowledge easily, unless that knowledge can be represented using additional attributes for each case in the dataset.

2.4.1.2. k-Nearest Neighbour

The k-Nearest Neighbour method is a Machine Learning algorithm based from Case Based Reasoning, for classifying cases based on their proximity in the problem space to other cases. That is, a case is given a classification based on how similar its values for certain attributes are to the values of those attributes for previously seen cases. The actual classification selected is the most common classification from the closest *k* cases. This algorithm does learn incrementally, potentially increasing its knowledge with each new case examined without requiring all data to be re-evaluated. However as the number of cases seen increases so too does the efficiency of the method substantially decrease (Kurniawati, Jin & Shepherd 1998). The method also lacks the capability to define complex relationships, except through storing a sufficiently large number of cases and examining a sufficiently large number of attributes that any relationships are found. However, as noted, this can cause the system to become highly inefficient. A further problem is that the knowledge gained from storing these classification boundaries is not easily viewable and understandable by a human expert, due to the potentially vast number and multidimensional nature of the spaces being defined (Hand & Vinciotti 2003; Kurniawati, Jin & Shepherd 1998).

2.4.1.3. Neural Networks

Neural Networks are an approach to learning how to perform tasks inspired by observations of dynamic learning networks in nature, for example, the neurons in the human brain. It is a purely automated approach in process, although it requires an initial specification of what a 'correct' result is for each case in order to begin learning. Although successes have been made, particularly for problems with noisy data that make human expertise difficult to apply (Mitchell 1997), neural networks are not suited to many domains. They are slow to train, do not learn well from complex data sets, and can not learn incrementally – extensive training is required to produce an accurate system. Most significantly, once the system has been trained, the knowledge it has learned is very difficult to review: the knowledge is stored implicitly within the configuration of the network, and there is no explicit domain modelling involved (Towell, G. G. & Shavlik 1993).

2.4.2. Machine Learning Drawbacks

Machine Learning methods are most effective in applications where the data that is being used for acquiring or discovering knowledge is sufficiently detailed that conclusions can be drawn from it alone, without further domain knowledge being applied – typically data that has been classified as being of a certain type, or that can easily be categorised according to type, allows statistical methods to find new relationships from the existing relationships and other data. The existing classifications represent a level of domain expertise that has been applied to the data, either from an expert who has examined each case and provided the classifications as extra information, or from an expert who knows which attributes of the set are important. Machine Learning methods are also only particularly effective in domains where the target knowledge (i.e. the knowledge the method is trying to discover) is relatively simplistic: complex relationships which have a practical use are difficult to derive without also deriving large amounts of other relationships which are meaningless, coincidental or overly specific to the dataset (Witten & Frank 2000). When the goal of the Machine Learning is Knowledge Discovery, not just Data Mining for the purposes of training an expert system, it is required for an expert in the domain to examine the relationships discovered and to determine what is useful and what is not (Abe & Yamaguchi 2005), particularly in domains such as medical

domains where performance can be critical. If the relationships are too many or too complex then this will be a highly difficult and time consuming process, negating the advantages of this approach. Another drawback is that Machine Learning can only discover knowledge that is present within the dataset being used: if the dataset is of insufficient size, or happens to contain statistical relationships which are not representative of the domain, then the method will either miss relationships or find misleading relationships; whereas an expert can use their extended knowledge of the domain to make judgements on what is likely to be coincidence and what is likely to be supported by further data (Hall & Smith 1998).

2.5. Combining Machine Learning and Knowledge Acquisition

Given that Machine Learning methods are impractical for analyzing data of a certain complexity without expert involvement, but purely expert-driven methods are too slow and do not provide the expert with the necessary assistance, a more expedient approach for these domains would be to develop an interactive tool that can help an expert to hypothesise about the information within a dataset, and prove or disprove those hypotheses based on the available evidence. The method should also be able to assist in the creation and validation of those hypotheses, based on the available domain knowledge and the domain data in the set.

Such a method would require that a level of domain knowledge be available, thus a Knowledge Acquisition phase would be required. Ideally, from the perspective of making the system easy to use for the expert and consistent in its approach, the Knowledge Acquisition technique used would be maintained throughout as the Knowledge Discovery method. This also has the benefit that any new knowledge found will be immediately integrated into the current model of the domain expertise. This is an important advantage, as it allows the Knowledge Discovery to be performed incrementally: newly learned knowledge can be immediately used to learn further. Therefore the Knowledge Acquisition phase of the method would carry on into Knowledge Discovery.

This is relatively simple to accomplish, by allowing the expert to perform Knowledge Discovery by deriving hypotheses about how the domain functions, based on relationships which are generated either by Machine Learning or by manual expert-driven analysis. The expert can then follow the usual Knowledge Acquisition steps as if they "know" the hypothesis to be correct: basing their justification on whatever information led to the hypothesis being created. If further domain model analysis tools are available, the results of adding this knowledge to the domain data can then be tested for validity. This does require however that the domain model be editable, as erroneous, unproven, or otherwise useless information will likely be added at various stages.

This approach is not suited to all domains. Domains without great volumes of data, or without complex target relationships, or domains in which the known expertise is relatively simple to model, will not benefit from this approach. This is because this method requires expert involvement at every stage, and can potentially take much more time than either a Machine Learning Technique or a statistical analysis tool. However, for domains in which the level of expertise required is too high for Machine Learning, and in which the volume and nature of the data is too complex for only human-based analysis, this hybrid approach should provide more relevant results in a more timely fashion.

2.6. A Complex Domain – Lung Function Analysis

In testing an implementation of the hybrid approach which has been discussed, a sufficiently complex domain is required. The following is a brief description of the domain of Lung Function, including why the domain is complex, the deficiencies in knowledge and research that make Knowledge Discovery a desirable activity and previous work that has been performed in the area.

2.6.1. Existing Knowledge

The domain of lung function, also described as pulmonary function, is a complex domain which is difficult to analyse. The purpose of the lungs is to oxygenate venous blood and remove carbon dioxide (Hughes & Empey 1981). Their effectiveness however is determined by several other components, including the airways, alveoli, pulmonary blood vessels, respiratory muscles and other respiratory controls (Ruppel 1994). Although the lungs perform very complex functions within the human body, they display few measurable outward signs of these functions (Laszlo 1994). Even those indicators which are apparent are difficult to measure effectively, due to the execution of the test interfering with the normal process of breathing (Hughes & Empey 1981; Ruppel 1994).

2.6.2. The Data

As the functions the lung performs are so diverse in nature, and in particular because they are measured by even more diverse means, no one test can provide a complete overview of all aspects of lung function (Hughes & Empey 1981; Miller 1987; Ruppel 1994). Although they each test different effects, using different means, all are essentially based on the same functions: this means that the information provided by these tests often overlap (Ruppel 1994). This further means that combining the results of many of these tests can produce much more detailed information about the patient's lungs' function than would be available by a single test. Also, due to the uncertainty within any medical domain, caused by the incomplete understanding of medicine and the complexities and wide degree of variation of the human body (Pribor 1989; Tsumoto 1998), any verification that can be provided from complimentary results from multiple tests will be beneficial in making conclusions with that data.

The tests themselves are divided into a number of distinct groups, with each group reporting on a specific aspect of lung function. Not all patients will have all tests carried out – the more complex tests are only performed if a medical practitioner deems them necessary in order to discover the cause of or to further define the symptoms a patient is displaying.

The first group of tests performed are Spirometry tests. These concern the volume change during specific breathing functions (Miller 1987), or in other words, the extent of the lungs' ability to move gas. Spirometry provides an example of where two results compliment each other to be more useful in analysis – the value FEV_1 (Forced Expiratory Volume in 1 second), while moderately useful on its own, is much more useful in conjunction with the test result of FVC (Forced Vital Capacity), as the FEV_1/FVC ratio can be used, with factors such as sex, age and height, to determine whether this function of the lungs is performing within a normal range (Ruppel 1994).

As it can be used as an indicator for asthma and COPD (Chronic Obstructive Pulmonary Disease; including emphysema and chronic bronchitis), both common respiratory illnesses, and also because the equipment is relatively cheap and easy to acquire, spirometry tests are the most common lung function tests performed (Ferguson et al. 2000).

The second group of tests are Lung Volumes tests, which attempt to measure the full capacity of the lungs; this is made difficult because the lungs will always hold gas that cannot be expelled. These tests are useful in identifying, clarifying or eliminating many dysfunctions or problems, both new and previously identified by other tests (Laszlo 1994; Miller 1987).

The final types of tests which make up the dataset are the Diffusing Capacity tests, which measure the ability of gas to diffuse throughout the patient's lungs and into their blood. This test covers very different functions of the lungs to the other tests, and so can detect specific types of problems. These tests can also be used in combination with previous results to determine specific illnesses, such as emphysema (Ruppel 1994).

These lung function tests generally do not provide enough information in themselves for a diagnosis to be made: they can give a measure of insight into the nature of the patient's lungs and how they may be functioning, and perhaps they can give an indication as to why the lungs are functioning as they are. However in order to complete a diagnosis the expert will generally require more detail, in the form of an

examination of patient history, a physical examination, chest radiography, blood tests, sputum examinations, and other tests (Hughes & Empey 1981; Miller 1987).

2.6.3. Data Mining in Lung Function

Perhaps because of the difficulty in obtaining lung function data, until recently there have been very few studies based on large bodies of test results. There have also not been any major compilations of test results into any one repository, except for the purposes of specific studies such as Oswald, Phelan et al. (1997) and Shaheen, Sterne et al. (1998), but these studies generally focus only on the results of a few of the tests and for specific purposes. It is only due to recent advances in lung function testing technology that many of the measurements have been able to be performed easily and commonly (Ferguson et al. 2000), and with further advances in technology in this area still being made, lung function data is being compiled now at a faster rate and in more detail than ever before (D P Johns 2006).

As has been noted, many of the test results interrelate with others in meaningful but complicated ways, and much greater knowledge can be derived by combining the results of different tests. However, these relationships and their implications have not been well explored. Generally it is the case that a hypothesis is formed, based on an interesting concept that has arisen from other work, and a specific study performed to ascertain the truth of the hypothesis (Aaron, Dales & Cardinal 1999; Glady et al. 2003; Punjabi 1998; Swanney et al. 2004). While this process is effective at discovering the validity of pre-existing hypotheses, little work has been performed in the discovery of these hypotheses, or in a simpler way of validating them.

The benefits of discovering a new relationship or use for a test result is difficult to quantify, as each may provide a wide range of advantages or none at all. These advantages may be financial, as in the case of Glady (2003), where an algorithm was developed which could, using only spirometry results, indicate to a high degree of accuracy whether the patient would require lung volume testing. Even though the algorithm was not always correct, the ability to be able to determine that further tests were not immediately necessary for even some patients saved the pulmonary function laboratory an estimated $20,000 (Canadian) per year. Not all discoveries

have benefits which are so apparent. However, any new information on testing lung function may be useful, for example to indicate a direction to examine in the future. Future technological advancements may provide a use for seemingly irrelevant information gathered, or the information may lead to a different means of performing a test.

Recently a great amount of pulmonary function test results have been compiled for examination for these purposes. A subset of this data was made available for the purposes of this development. The data was collected by the Respiratory Laboratory, Royal Perth Hospital, Western Australia. It consists of the test results of 484 adult Caucasian subjects who had full lung function tests performed, on the recommendation of a doctor, consisting of full spirometry, measurement of lung volumes and diffusing capacity. Therefore the data is a representative sample of patients who have been referred for testing – it is not representative of the lung function of the general population, but nor does every patient necessarily have significant dysfunction with their lungs.

Given the current incomplete knowledge of the way the test results interrelate, and the potentially very pertinent new information about patients' lung function that can be derived using these test results, and given that we now have a much greater volume of data than before, the domain should be able to benefit greatly from the application of data mining techniques to the gathered data.

Some data mining work attempting to model expert knowledge has been performed in this domain. Tsumoto (2004) described a method to automatically acquire medical knowledge and model it as rules, using rough sets, and it has had success. Singh (2006) implemented a system for automated medical annotation of databases of lung images. The system also includes a Machine Learning algorithm to automatically induce rules to make these annotations, showing that automated Machine Learning has been successfully implemented in a medical domain.

There have also been several successful expert systems developed for medical domains. The increased demand for accurate and timely recommendations and diagnoses, and the importance of getting those recommendations and diagnoses

correct, as in any medical domain (Pribor 1989), has led to a desirable state for expert system development. In the field of lung function, Pulmonary Consult is an expert system for interpreting lung function results, which has been developed and made commercially available (Snow et al. 1988), and is available now (*MedGraphics Pulmonary Consult™ Software* 2006). The continued use of this system shows that expert systems can function within the lung function domain, and therefore, that expertise within this domain can be extracted and modelled effectively.

3. Methodology

3.1. Introduction

To test the concept of the hybrid Knowledge Acquisition and Machine Learning technique for Knowledge Discovery required the implementation of a system and the testing of that system over a dataset. The testing of the implementation will be discussed at a later juncture. This section will deal with the implementation of the method.

Of the Knowledge Acquisition methods examined, the MCRDR algorithm offers the most beneficial framework for implementing this hybrid method. In particular the fact that the Knowledge Acquisition process is intuitive and no knowledge engineer is required, but also that the knowledge base is readable; new knowledge is validated as it is acquired; the learning is incremental; and the process is entirely expert driven. Some modifications and additions are necessary to facilitate the Knowledge Discovery process, and to assist the expert in finding new knowledge. This section contains a description of a new Knowledge Discovery method based on MCRDR Knowledge Acquisition, followed by a summary of the implementation.

3.2. Exposed MCRDR

There are a number of differences between regular MCRDR and the proposed Exposed MCRDR (EMCRDR). There are features which have been added, and many modifications made to the existing process. A discussion of why changes have been made and their implications will follow later in the study.

3.2.1. Method Description

The EMCRDR method, in general, follows the same pattern as the MCRDR Knowledge Acquisition cycle: the expert takes a case, classifies the case according to their understanding of the domain, and runs the case through the system's inference procedure to compare the system's classifications with their own classifications. When the expert determines that a misclassification has been made or a classification missed, the expert changes the appropriate classification or adds the new classification, and creates a rule justifying why this classification should have been made.

The method can be divided into two phases, the Knowledge Acquisition phase and the Knowledge Discovery phase, although there is very little separating the two and no requirement that they be separate. For the Knowledge Discovery phase to work to its best potential a sufficient amount of Knowledge Acquisition should be performed first, to attempt to model enough of the known domain knowledge to build upon in the Knowledge Discovery phase.

3.2.1.1. Knowledge Acquisition

This phase involves trying to build up a sufficient knowledge base that there is enough background information to be able to determine new knowledge about the domain. The process is generally cyclic, as with regular MCRDR Knowledge Acquisition, but with other options for assisting in the knowledge base development. The main cyclic approach is for the expert to examine a case from the dataset to find all the classifications for that case. The case is then run through an inference using the system's knowledge base, exactly as with normal MCRDR. The expert examines the classifications returned by the system and compares them with the manually made classifications. Any discrepancy that is found, assuming that the expert has not made a mistake – this would be realised by the expert when the classifications are found to not match – progresses to an explicit Knowledge Acquisition step for defining a rule.

Firstly the expert selects the classification that should have been made (or "No Conclusion" if the expert is removing an erroneous classification – this is named a stopping rule), or defines a new classification if required. The rule definition then

proceeds, with the expert asked to justify why this classification should have been made by defining rule conditions according to the reasoning process the expert performed in determining that the classification was incorrect (or missing). This is where one of the first divergences from regular MCRDR Knowledge Acquisition is implemented: the regular cornerstone-case based rule validation is replaced by a dataset-based rule validation. When a rule condition is defined and the expert performs rule validation, the method returns a list of all the cases from the dataset which are now satisfied by the rule. The expert then examines these cases to determine if any should not be covered by the rule, and adjusts the rule conditions accordingly if required. Once the expert is satisfied that the rule is correct the rule is saved to the knowledge base. The position in which the rule is added to the knowledge base depends on the purpose for which the rule was created: if added as a new classification for a case, the rule is added at the top level of the tree, beneath the root; if added to remove or change a previous classification, the rule is added as a direct child of the rule(s) that caused that classification (the exact rules are selectable by the expert when stating which classifications are incorrect). Once the rule is added the expert returns to the beginning of the cycle and can select another case or use the other option, viewing the knowledge base.

3.2.1.2. Viewing the Knowledge Base

Viewing the knowledge base is an integral part of the Exposed MCRDR method, and is the primary feature from which the method draws its name. As well as allowing the user to view the dataset and to define rules based on those cases, EMCRDR allows the expert to view the knowledge base, and select individual rules to edit or delete. The functionality provided by this could be considered to be a third phase to the EMCRDR method, in that the approach is quite different. However, this functionality can be used at any and every stage of performing Knowledge Acquisition or Knowledge Discovery. This functionality is required in order to allow the expert the ability to gain an understanding of the domain knowledge that has been recorded at any stage, and to allow the expert to make guided decisions as to how to edit or add to that knowledge productively.

The ability to delete a rule has two further options – to also delete the rule's children or to keep them. If kept, the sub-tree(s) of the children are moved into the deleted

rule's place, and in order to maintain the context-sensitive nature of the data, the deleted rule's conditions are each added to the conditions of its first level children rules.

Editing rules provides identical functionality to that of rule definition, except that a number of conditions are already defined. The validation process upon any modifications remains the same.

3.2.1.3. Knowledge Discovery

The other aspect of the EMCRDR method is that of Knowledge Discovery, the functionality explicitly designed to facilitate discovering new knowledge about the domain. This goal is achieved, again, via mostly the same method as the Knowledge Acquisition: defining a rule based on a certain case which is representative of the classification being considered. This provides the advantages of having a fully incremental discovery process, with validation and verification of any new knowledge as it is discovered. However, the difference provided by the discovery phase is in the assistance given to discovering trends and creating rules which might show interesting features of the domain. These two features are combined into one tool which is integrated into the rule definition functionality of the method. The tool provides the ability to subdivide the dataset according to attributes selected by the expert. The tool works on a subset of the dataset, specified by the rule currently being worked upon, and allows the expert to further subdivide that set, to either explore attribute relationships or to generate rule conditions.

From the initial set (the "excluded" set) the expert can move cases, individually or in groups, to the "included" set. The method then calculates the range (minimum and maximum) of the values for each attribute, from the set of "included" cases, by iterating through each attribute and each case. These ranges can then be selected for consideration, which will move all the other cases in the "excluded" set which are within that range to the "included" set. Once the expert is satisfied that the "included" set consists of all the cases which should have the classification that the expert is trying to define, a rule can be created based on the attributes of the selected ranges. This rule will be in the form (given that the range for Attribute A is selected and is 0 - 0.5, and the range for Attribute C is selected and is 0.01 – 0.02): IF A > 0

AND A < 0.5 AND C > 0.01 AND C < 0.02 THEN <*Classification*>. This generated rule can then be freely edited via the normal rule edit processes, in order to adjust the rule to be more generally applicable to the domain or more accurate to the relationship the expert is testing.

3.3. Implementation

To test this Exposed MCRDR method requires the implementation of a full MCRDR system, with modifications and enhancements made to the system interface, the inferencing engine, and the handling of the knowledge base. The system was implemented as a Windows Application and the MCRDR engine as a series of Dynamic Link Libraries (DLLs) using the .NET 2.0 Framework, with the domain data and knowledge base stored in a MySql 5.0 database. These tools were chosen as both are readily available and well supported technologies.

The system has been implemented to be able to work for only a single user, rather than many experts inputting their expertise, although it has been designed for the possibility of future expandability in that regard.

3.3.1. Domain Modeling

In abstract terms the goal of all Knowledge Acquisition is to find a way of accurately modeling a domain (Liou 1990). We attain and structure the domain knowledge using the MCRDR algorithm. However, in order to store this knowledge, we must also find an adequate data model to represent the various elements of the domain, including allowances for how they interrelate. We also must consider how this data will be stored.

3.3.1.1. The Database

The main concern in modeling the domain for this project is finding a means of representing and storing the cases that make up the dataset, and the rules that are based on those cases. We then have to consider any subsequently generated conclusions, and any other metadata we may require to allow us to work with the

data as we would like to, including the potential for future work. The core of this is representing the cases themselves. This was mainly a simple process as the dataset we were using was provided in a flat Excel spreadsheet format, and so the majority of attributes could simply be transposed directly into a table, with each column in the spreadsheet a column in the table and each row in the spreadsheet a record. The data types of these attributes were similarly simple, as most of the attributes are numerical values to 2 decimal places. A more unusual aspect of the data in the spreadsheet is that many of the columns (41 of the 101) are in fact formulae. A set of these derived attributes (21 of the 101), labeled 'reference' attributes, were calculated based on other attributes that are actual measurements taken from the patient in various tests. These reference attributes represent the predicted result for other tests. Many of the other derived attributes (19 attributes) are calculated from the actual test results divided by the predicted test results (giving a "percentage predicted" attribute). The remainder of the derived attributes are calculations based on already known relations between attributes, such as one attribute subtracted from another, or divided by another. All of these derived attributes are not explicitly stored in the database, as they can be calculated in the intermediary object-oriented data model used between the database and system program.

Although the dataset does not contain information identifying patients, the possibility was raised of extending the system in the future to take into account attribute change over time for a single patient, as used in the system PEIRS (Compton & Edwards 1994). To this end, we separated the data set (in the database and data model) to consist of a "patient" and a set of "test results". The patient object includes the unchanging patient-specific data we have available in ethnicity, sex, and year of birth. The test results object then contains all other attributes plus a patient ID to link it to a patient, and the year the test was performed to calculate the patient's age at the time of the test. Although the test results are categorized into groups in the interface, based on the actual tests performed and meaning of the data gathered, all are stored in a single table in the database for simplicity and as each attribute is treated as an equal at this stage.

3.3.1.2. Data Types

The majority (95 of the 101) of the attributes used in the dataset, including all the derived attributes, are real numbers. A distinct few (3) are whole integer numbers and have been stored as such. The only complications in the data types are the Smoker, Ethnicity, and Sex. An Enumeration has been defined for each of these within the C# data model which restricts the values that can be entered into these fields. Smoker has been restricted to "Y" for yes; "N" for no; and "X" for ex-smoker. Ethnicity is "W" for Caucasian; "O" for other, although the set only contains "W". Sex is simply "F" or "M". Each of these attributes has also had the value "MISSING" added as a possibility, although no values in the dataset are missing, again for future expandability. The use of "missing" as a distinct value is discussed in the Section 7 **-** Further Work.

3.3.2. The Interface

Much of the strength of the MCRDR Knowledge Acquisition process arises from presenting the user with an intuitive means of updating the knowledge base, and hiding the complexity of how that knowledge is actually structured (Compton & Jansen 1989; Richards & Busch 2003). For example, the typical MCRDR Knowledge Acquisition process is designed to operate under circumstances where any domain expert can use the system, and add knowledge as appropriate, without any understanding of the content and structure of the existing knowledge base. The expert views the classifications that the system currently gives, and simply chooses to add to, change one or more, or remove one or more of those classifications. The expert then justifies this decision, in the form of a rule. Hence, a user can perform Knowledge Acquisition within the system without necessarily being aware that the knowledge base is structured as a tree; the user simply understands that a new rule can supersede other rules, or can be considered in addition to other rules. Traditionally, the method has been deliberately implemented in this manner in order to provide a natural and intuitive experience to the user, without requiring them to understand the knowledge base tree structure, or understand the complexities of how the system produces the results it does (Compton & Jansen 1989; Kang, Compton & Preston 1995). As such, much of the strength of MCRDR Knowledge Acquisition over other expert system Knowledge Acquisition methods is derived from having an

effective interface, and so the development of the interface needs to be considered in equal regard to the development of the engine.

3.3.3. The Rules

The rules in the system each consist of a number of conditions, or clauses, and a classification, or conclusion. Each condition consists of an attribute, operator and value set. The attribute is selected from a list of all attributes in the dataset. The operator is selected from the set of: '= =', representing just 'equals'; '!=' representing 'is not equal to'; '>=' for 'greater than or equal to'; '<=' for 'less than or equal to'; and '>' and '<' for 'greater than' and 'less than' respectively. The inequality operator was included as domain experts make extensive use of "negative knowledge" in justifying their conclusions, as in most medical domains (Tsumoto 2000). The value is any value entered into the field by the expert, defaulting with the value for the selected attribute for the current case (the typical cornerstone case, which the rule is being based from).

The final component of a rule, the classification, consists of a title and a description, which are defined before the rule is defined, and the correct conclusion selected from the provided list of all conclusions.

3.3.4. MCRDR Modifications

However, the requirements of Knowledge Acquisition for data mining purposes are different to those of Knowledge Acquisition for the training of an expert system. There are a number of modifications to the standard model which needed to be made, and a number of additional features which needed to be added. We will now consider the features that were modified or added, why they were necessary and how they were implemented. A discussion of the advantages and disadvantages of each of these changes will be considered later in the Results and Discussion section, in association with the results obtained.

3.3.4.1. Using a Dataset

The most significant change to the design and function of the system is the inclusion of a dataset. In regular MCRDR, cases are presented one at a time to the inferencing

engine, and the results returned for that case. If the expert determines that the knowledge of the system needs to be updated, the new knowledge defined is based from the case that gave rise to the mistake, and that case is stored with the knowledge base as reference (and called a cornerstone case). However, as the goal of the system is to take a given set of cases and discover interesting features and patterns from that set, the system has an initial range of cases with which to work. It is necessary to base the Knowledge Acquisition from this initial set of cases, rather than case-by-case as they are presented to the system, for a number of reasons. Firstly it makes the validation of rules much more effective and efficient, by providing a large range of cases to be considered and compared from the beginning. A case-by-case approach would require a significant amount of time to build a large enough set of cornerstone cases to be able to validate rules effectively, as it would potentially (particularly early on in development) invalidate every rule when each new case is added. Hence although the case-by-case approach would run the validation process faster initially, it is at that time a less effective process which is potentially redone every time a new case is added.

Using a dataset also provides a definite scope for the discovery process, as opposed to an open-ended approach of adding as many cases as possible over time. This restriction can allow the user to focus on specific types of cases, all at once, rather than waiting for such cases to occur in the normal use of the system. Similarly, if the dataset can be said – or influenced - to contain a range of domain representative cases, it allows for a more complete analysis of the domain rather than a purely incremental approach which cannot say when adequate domain knowledge is reached, other than by analyzing the accuracy of the system over time and seeing when the system error rate drops below a predetermined threshold.

Having a set of data from which to base rule validation also subtly changes the implementation of the MCRDR Knowledge Acquisition process, and more significantly changes the interface. The algorithm itself performs rule validation by adding the new rule to the knowledge base and then running every case in the dataset through the inferencing process to find all those cases which match the new rule. This process can be optimized by only inferencing those cases which already satisfied the parent rule of the new rule being created, as only those cases can

possibly reach the new rule's position in the tree, and so all other cases are automatically known to fail to satisfy the new rule. The system uses all dataset cases (rather than only those already used as cornerstones) in order to take advantage of the availability of the wider set from the beginning, for the reasons noted above.

However, using the entire dataset as validation from the beginning requires a redesign of the use of cornerstone cases. As the set of cases being used is potentially very large, storing cornerstone cases in the typical sense becomes impractical. Whereas most MCRDR systems would, upon the creation of a new rule, make a copy of the case used and store this as a cornerstone case for that rule, as we already store all the cases the system need only store the relationships between each rule and the cases it covers. As such, the system does not explicitly have cornerstone cases, but rather just stores the list of cases that are currently covered by each rule, and updates this is dynamically as required. The rule creation does require the concept of a cornerstone case, as the new rule conditions need to be based from the case which caused the expert to reconsider the validity of the knowledge base (i.e. the new rule should cover the case which caused the system to provide an incorrect classification to provide an extra measure of validation). As with previous MCRDR systems it would be potentially beneficial to store a copy of each case as it is used as a cornerstone, so that were the case ever to be modified the origins of the rule would still be available upon request (Compton & Jansen 1989). However, as we are storing every case that matches the rule upon validation at each step, without distinction, any of these cases will be a valid cornerstone for the rule. Because the rules are being created from a dataset, always in the context that the rule covers a specified set of cases, once the rule is satisfactorily defined it is irrelevant which case originally caused the rule to be made.

3.3.4.2. Interface Modifications

The interface modifications are more significant. Although there are large differences in the general appearance and contents of the interface in order to present the dataset to the user in an easily readable and browseable format, the most important factor is the difference that presenting this information makes to the way in which the user can use the system. Typical MCRDR rule validation presents the user with the previous cornerstone cases which are also covered by the rule, and offers the option

of accepting this new or modified classification as valid for each particular cornerstone case, or rejecting it, which requires a justification – in the form of a difference – to further refine the new rule so as not to include the rejected case. However, the data mining system presents all the cases in the dataset which were covered by the new rule, requiring and allowing the user to take a much more holistic view of what the rule they are defining means in relation to the domain, rather than performing a purely case-by-case analysis. Most significantly, it allows a user to define a rule by examining the entire dataset and developing a rule based on how the rule subdivides that set, rather than purely basing a new rule on their understanding of the domain in relation to the case being considered. This ability is the essential element that changes the process from being a Knowledge Acquisition process into a data mining process.

3.3.4.3. Viewing the Knowledge Base

Another modification that was considered likely to be beneficial is to allow the data mining user to view the structure and content of the knowledge base, in order to gain a more holistic understanding of the knowledge they have acquired, rather than having all the knowledge encoded and accessible only through a case-by-case classification. Viewing the knowledge base in an explicit format also allows the expert to see each rule in the full context within which it was defined: a very important consideration when attempting to consider the full scope of the knowledge gained, and an aspect which is not apparent in typical MCRDR (Compton & Jansen 1989). The system implements this via a visual tree structure, explicitly describing the knowledge base in the same way that the rules are logically linked together. This approach was chosen over other representations, such as hierarchical grids, or a flat grid, because it is the best way found to provide a clear explanation of rule context in regards to the entire knowledge base. It also presents a way to easily show a broad view of the knowledge base in one screen and, through the ability to minimise or maximise branches of the tree it allows for the ability to narrow down on smaller subsections of the larger knowledge base.

3.3.5. Additional Features

While the MCRDR implementation required many modifications for it to be applicable as a data mining tool, it also required a number of additional features. These features are enhancements to the standard MCRDR method, some of which may be applicable to other MCRDR applications, and some of which are focused more specifically on data mining applications.

3.3.5.1. Deleting Rules

The first non-typical enhancement to be added to the MCRDR system is an implementation of rule deletion. Most MCRDR systems don't allow explicit deletion of rules, instead making use of stopping rules that have the same conditions as the parent rule, to effectively remove a rule from the knowledge base (Kang, Compton & Preston 1995). This has been done to avoid the complexities and dangers inherent in removing a node within a tree – when to reconnect the children, and if they are reconnected, how to handle the invalidation of the context of their rules – and is the primary source for the MCRDR ideology of never deleting data (Compton & Jansen 1989; Kang, Compton & Preston 1995, 1998; Richards & Busch 2003).

However, in the data mining context – of trying to discover new knowledge, by creating test rules and observing how they modify the results obtained – there are potentially many cases where the user will wish to delete rules previously created. Although there is technically no case of rule deletion which a stopping rule implementation cannot replicate, because the user is allowed to view the knowledge base itself and because it is expected that the user will create many rules which they will wish to delete, it will was anticipated that it would provide the user with a much clearer view of the knowledge base. This is an important consideration as the user will always be a domain expert and therefore cannot necessarily be expected to be adept at understanding the MCRDR tree structure of knowledge.

The rule deletion was implemented to be as generally applicable as possible, as there is currently little data about how an expert will use the rule deletion, particularly as related to the domain of lung function and the application of data mining. There are two cases to choose from when deleting a rule, selected by the user when they are

performing the deletion: to remove the rule and all its children, or to only remove the current node and keep its children. The former case is the easiest to handle, by simply removing that node from the tree, and thereby disconnecting all the children nodes, making them impossible to be reached by traversal. The latter case is complicated by the requirement of maintaining context within the tree – a rule is created entirely within the context of its parent, and to remove the parent would potentially open the rule to cover a range of other cases that were not intended to be covered. To handle this, when deleting a rule and selecting to keep the children rules, the conditions contained within the rule being deleted are copied into all of the first level child rules. This ensures that all the children rules will produce exactly the same classifications as before the deletion, while removing the classifications provided by the rule that was deleted. It is a possibility that the expert may actually desire that the child rules' scope be expanded to include those cases which will now not be excluded by the parent rule (although, if this were to occur, it is likely that the child rules should actually have been included as siblings of their parent). In a typical MCRDR system the expert would be required to remove the children as well, and redefine their rules to not include the deleted rule's conditions. While it would be possible to present the expert with an option of carrying down the deleted rule's conditions or not, this would potentially be a difficult concept to communicate the consequences of effectively. However the system also contains the option of editing rules, so in the situation mentioned the expert would be able to modify the children rules and remove those clauses just added.

Although rule deletion has been implemented, the system still does include an implementation of stopping rules – and indeed the system's implementation can be used in such a way as to never need to use the rule deletion option, although that is not a preferable outcome. Stopping rules are added whenever the expert selects the option to remove a classification from the results screen – this is primarily to allow the expert to define a selective stopping rule, in which only a subset of the parent rules cases satisfy the stopping rule (Bindoff 2005; Kang, Compton & Preston 1995). In this situation those cases do not receive a classification from that branch of the knowledge base, whereas certain other cases still may. Explicit rule deletion can only be performed by viewing the knowledge base and selecting the rule to be deleted, although the expert can still achieve the same result by defining a complete stopping

rule which excludes every case covered by the parent rule. The explicit rule deletion allows the expert to consider the knowledge base and find rules (or branches) which serve no purpose, which are deemed irrelevant or incorrect, or which were added to attempt to prove a hypothesis that failed, and remove them.

3.3.5.2. Editing Rules

Another addition to MCRDR within the system, which again defies the standard MCRDR approach, is the editing of existing rules (Compton et al. 1992; Compton & Jansen 1989; Kang, Compton & Preston 1995, 1998; Richards & Busch 2003). This feature allows the user to view existing rules and modify their conditions or conclusion.

The editing of rules is in contradiction to the suggestion by Compton, Cao et al. that implicit editing of the knowledge base has no inherent advantage and only risks introducing errors, as "the circumstances in which knowledge is appropriate are never fully defined….[meaning] the fix will never be complete" (Compton, Cao & Kerr 2004). For this reason, in a regular MCRDR system, modifying rules explicitly is never performed; every situation where the user wishes to wishes to modify a rule involves adding a new child rule (Compton & Jansen 1989; Compton et al. 1993; Kang, Compton & Preston 1995, 1998). However, Compton, Cao et al. did note that while an RDR approach does seem to provide an easier method of Knowledge Acquisition than free editing, they were unable to prove the conjecture at that stage (Compton, Cao & Kerr 2004).

As this method seeks to discover new knowledge and to store it in a readable and useable format, it is suggested that editing rules does have an inherent advantage: that of being able to change the knowledge base to have it conform to an expert-comprehendible view. Also, as with the removal of rules, it is expected that the expert will commonly be testing new rules – of which they are unsure of the validity – and so will frequently want to perform rule modifications to refine their current hypothesis. It is suggested that if the expert can view the full effects that editing a particular rule has on the knowledge base, and if the expert has the ability to reverse any modifications they have made, the expert's understanding of the domain should

restrict introduced errors to an acceptable minimum: because they will be able to "undo" any mistakes they have introduced.

Allowing rule editing to be initiated from looking at the knowledge base also allows the expert to tidy the knowledge base if it becomes convoluted and contains contradictory, unnecessary rules. This is a required feature if the expert is to be able to easily examine and interpret the knowledge base. As outlined in the description of the rule removal functionality, rule editing is also useful in the case that the user wishes to refine existing rules whose meaning has been modified by rule deletion, which enhances the applicability of both features. The disadvantage being that the knowledge can be impaired by editing without a full understanding of the context (which may be impossible to obtain)

The editing process is performed by taking the user back to the rule creation screen, except with the conditions grid already containing the existing conditions and the existing conclusion already selected. All other features of the define rule screen are present and enabled.

A consideration when implementing rule editing, if implemented as a replica of rule creation, is which case to use as the cornerstone case. To address this concern, a system was implemented to generate a cornerstone cases list. Upon validation and saving of a rule, the system would identify and store all cases in the dataset which match the modified rule. Once this has been performed we do not keep track of which of these cases was the original cornerstone case (that the rule was founded from), because as we are basing the Knowledge Acquisition from a dataset rather than a case-by-case appraisal, it is considered that the rule was created to represent all those cases. Therefore, when a rule is being modified, we consider the base cornerstone to simply be the first cornerstone returned by the list. However as the user may wish to modify the rule to not include whichever case is returned as the first case, we do not strictly enforce that the rule covers this case – we provide the user with a warning if the rule does not cover this case, but still allow the modifications they have made to be saved. A discussion of the potential impacts of this decision is provided in the Results and Discussion section.

The most significant problem with implementing rule editing is handling how rule modification will affect the results provided by the children of that rule. In the system we have not implemented a programmatic means of handling this situation. It is assumed that the expert will be familiar enough with the knowledge base, through the extended detail provided by the system and through explanations provided by the author, that they will understand the implications that modifying a rule will have for that rule's children. We also consider that it is usually the case that a rule modification will simply be a clarification of one or more existing conditions, and that clarifying a rule in this way will not result in misclassifying cases with the child rules, as the child rules were intended to only have been considering those cases now covered by the modified rule. The worst case allowable by rule editing is if the expert completely recreates an existing rule to perform a totally different classification, thereby likely making the child rules completely irrelevant and erroneous. To avoid this is it is suggested to the expert that in the situation where they do wish to remove an existing rule and then add a new rule that they go through the processes of rule deletion and rule creation, rather than the shortcut of editing. However it is also assumed that in their reviews of the knowledge base the expert will be able to see any erroneous rules or rule branches and remove them at that time, and so this is not considered to be a significant problem beyond an overall loss of efficiency.

3.3.5.3. Validation

To facilitate a data mining approach, it is also necessary to modify the format in which validation is presented. If a newly defined rule covers a case that had previously been inferenced, and the expert had not at that time determined that a classification was missing, unlike a typical MCRDR system this does not necessarily imply that the expert has missed a classification – the expert may be simply attempting to describe the case in a different manner, or using a different approach, than previously. This also results from the presence of the dataset in the system and the relaxing of the concept of cornerstone cases, since in order to be able to determine that a new rule may conflict with a previous rule for a given case, the system needs a means of knowing whether a case has been considered to be completely classified. This is a process usually performed using cornerstone cases. The system could store a value for each case representing whether the case is

considered to be completely classified or not, but this is contrary to the goals of the system: the expert does not know when a case is completely classified, as they are attempting to find new classifications and ways of making those classifications.

With this in mind, rule validation in this system does not present to the user a list of all the cases that are affected by the new rule and have previously been inferenced, requesting confirmation from the expert that all the classification are correct. Instead a list of all classifications for each case is provided, so that the expert can determine for themselves whether each case has a classification which conflicts with the new classification (for example if a case is already classified as having severe airflow limitation and is now being classified as mild airflow limitation), and allow the expert to modify the rule appropriately.

3.3.6. Data Mining Features

Perhaps the most significant additional feature implemented is a tool, integrated into the standard system, which assists in the generation of rule conditions based on the similarities between attributes across cases in the selected sub-dataset. This feature was determined to be particularly relevant in the system as one of the goals is to attempt to determine previously unknown correlations between attributes or between cases. This tool is also relevant in any MCRDR system where the user does not have a complete knowledge of the domain and wants to examine exactly which attributes and what values might be most profitably used, or for a user who wishes to explore the workings of the domain beyond their normal understanding.

This functionality is accessed from the rule definition (or edit) screen. It examines only those cases that are currently covered by the rule, as if the rule had already been added to the tree (and so is inclusive of the restrictions imposed by the parent rules). The result of this is that if the user were to enter this tool before defining any conditions for a rule, the set of cases examined would be all those cases which are covered by the parent rule (i.e. all those cases which would reach this point in the tree). These cases are presented in a list (‘excluded cases’), from which the expert selects cases which they determine should be covered by the current rule and move into the ‘included cases’ list. Every time the ‘included cases’ list is updated, the

system recalculates the current value range for each case attribute, and displays these to the user (i.e. for each attribute, it takes the minimum and maximum value present in the list of included cases). Each of these are presented along with the count of the number of cases in the 'excluded cases' whose value for that attribute also falls within this range.

Selecting an attribute's range highlights all those cases in the excluded list which are also covered by that range. Choosing to use a range, by ticking the appropriate checkbox, moves all those cases covered by that range to within the 'included cases' list, and recalculates all the ranges that have not been selected. This last point is an important element of the implementation if more than one attribute is required to determine the correct set of cases to be covered.

Once a range has been selected to be used, it is locked with those values and will not be updated when the user changes the composition of the two case lists. If it is ever deselected, then all those cases which were moved when that range was selected will be moved back to the excluded case list. This makes the order in which the attribute ranges are selected have an influence on the end result; as selecting a range updates all the other ranges.

Implementing the method in this format provides a greater degree of control to the user over which cases are included and when, as it does not restrict the included cases to being based on strict ranges. However, this also means that the method relies on the user having a level of expertise within the domain, and particularly on the user having the ability to identify which attributes are more likely to be useful, and in which order.

The goal when using the tool to generate rule conditions is to have all cases which should be covered by the rule in the 'included cases' list, and all other cases in the 'excluded cases' list. Once this is achieved, the user clicks the 'confirm' button to add the currently selected ranges into the rule as conditions. It will usually be necessary for the expert to then modify the values within these conditions to be more generalized and less case-specific, assuming these rules are not meant to be entirely dataset specific.

In addition to providing a simple method of generating conditions for rules, this tool has the potential to be useful for more basic exploration of the relationships present within the dataset. If used in conjunction with an already defined rule, the method will show the expert any attributes which perform the same function as those which have been used to restrict the set so far – that is, once the attributes that define the rule have been included, any other attributes which have a very low count of cases which they further cover could be said to be a possible alternative to those attributes which were previously used in the rule, and so be worthy of further examination. Although a low count certainly does not guarantee a similarity of applicability, it is an indicator that it may be worth examining further, which matches the goals of the system. In the same manner, attributes whose count is high can be reasonably assumed to be irrelevant to certain applications or conclusions.

Further data mining and analysis functionality is provided for by allowing the expert to export any set of cases which are satisfied for a certain rule or defined within the data mining tool. The cases are saved, either with their classifications or without, as a Comma Separated Value (CSV) file, which can be opened by most spreadsheet programs for extended manual manipulation and plotting.

3.4. Method Evaluation

3.4.1. Testing Process

In order to test the system a leading expert in the field of Lung Function was given the system, with a dataset of 484 cases, for a period of three weeks (25 days) to firstly build the knowledge base, and secondly attempt to discover new knowledge about the domain. The expert was given a brief explanation of how the knowledge base was constructed and how the classification algorithm functioned. The expert was then instructed about how to define rules in the system by the usual MCRDR method of analysing a case, manually finding the appropriate classifications, and then adding, changing or removing the classification results that the system provided for the case until the system's responses match the expert's. The expert was also

shown how to define and test a rule based on stand-alone knowledge rather than based entirely from a case. When the development of the knowledge base was approaching completion the expert was shown how to use the 'Similarities' screen to compare the attributes within subsets of the data and to automatically generate conditions for a new rule.

3.4.2. The Dataset

The dataset which was used is a subset of 484 patient results from the collected lung function data. Although the system is designed to examine much larger datasets, and is scalable to larger numbers of cases, this set was used as it had already been manually 'cleaned' by the expert to remove any incomplete cases (those that have missing values) or are particularly noisy (have clearly erroneous values). These considerations reduce the potential impact of noise and ambiguity factors in the study and so make it more likely that we are accurately representing the ability of the method within the domain. The set does not contain any personally identifying attributes, and no one patient is represented by more than once case.

3.4.3. Evaluation of Testing

To analyse the success of the method and understand the implications for future research and development, both with the MCRDR algorithm and in the lung function domain, a number of factors must be examined: how the expert used the system, how difficult to use the system was, the knowledge base that was developed, how that knowledge base was developed, the length of time taken to derive information, and how influential the resultant "mined" information is to the domain.

3.4.3.1. Usage Logs

In order to look at this information the implementation includes automatic logging features for recording each action that the user makes in using the system. To facilitate the logging of deletions meaningfully, no records are ever deleted from the database, and are instead only marked as deleted or not. This is acceptable for the system as it stands as a prototype using a limited dataset over a limited timeframe, but if the system is to be expanded in the future and used more routinely then this design would likely need to be reconsidered. To maintain meaningfulness of edit

actions, the logs also store the values before and after any edit. This should allow a full understanding of what the content of the knowledge base was at any given time of development. The actions that were logged can be found in 9**.** Appendix A – System Usage Logs**.**

4. Results

To examine the success of the method, and what the impacts of this method are on the field of Knowledge Discovery, a number of analyses can be performed. Firstly, a direct examination of the characteristics of the knowledge base produced by the testing. Secondly, an assessment of how the system was used, via the previously mentioned usage logs. Thirdly, an examination of a review of the system, written by the expert, describing the new knowledge obtained, the effectiveness of the system, and areas in which the system might be improved.

4.1. Knowledge Base Examination

The first and possibly most significant result to be found by this study lies in the structure of the final knowledge base. The tree that was produced in fact only consisted of a single level beneath the root node – one level of rules all at an equal level, with no child rules. There are various possible reasons for this. The reason with the most support of the evidence is that the expert appears to have not wanted to define child rules at any stage, because the expert did not fully understand the process (D P Johns 2006). The expert would instead define more detailed rules to cover all eventualities, and further new rules to cover any complexities that might be involved in the area. This theory is also supported when the number of conditions per rule is examined: as shown in Figure 2 nearly half of the rules in the system consist of over 2 conditions; also the average of 2.4 conditions per rule is significantly higher than the 1.7 found by Bindoff (2005), and approximately 1.5 found by Compton and Edwards (1994). It is worth noting that the knowledge base for this system is still simpler than the knowledge base from Compton and Edwards' study, in which the average number of conditions satisfied in finding a classification was typically 5, approximately double that of this system.

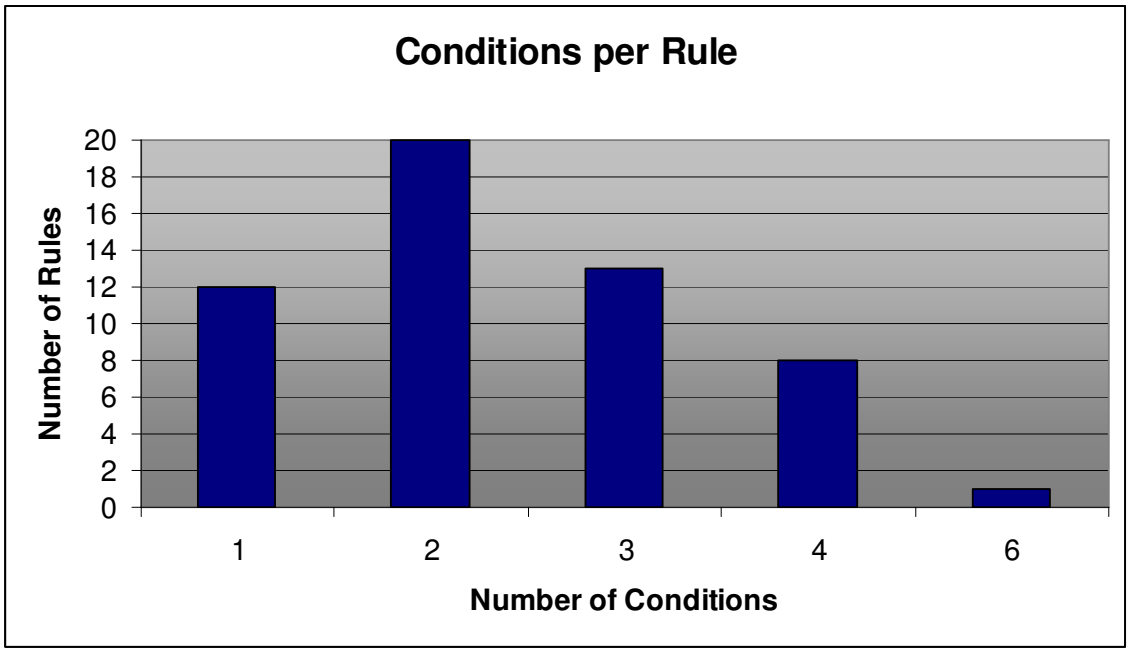

Figure 2: Number of Conditions per Rule, showing an average of 2.4412

It is suggested that another reason for the single-level tree structure of knowledge base is that it is simply more easily understandable for the expert, particularly in the context of data mining. As the focus for the data mining is to produce discrete new 'pieces' of information, in the form of rules, it follows that the expert would be considering the domain knowledge – or at least domain knowledge representation – in this manner, and so will attempt to devise rules in this style, rather than follow the normal process of 'building' the knowledge base case-by-case via correcting the system's classifications.

4.2. System Evaluation Tests

4.2.1. Classification Accuracy

A number of specific tests have been used to describe the success of the system, as follows. However it is worth nothing that many of the normal evaluations of classification accuracy are not relevant for this system, as the system is focusing on rule generation for data mining purposes, as well as the generation of an accurate expert system. An overall accuracy measure for the classifications found cannot be derived as there is no authority on what the final set of classifications for any given

case should be: it is assumed that the expert will be correct in his own classifications of the cases, and hence once he is satisfied with the system's responses, that the rules in the knowledge base will therefore be correct (excepting human error, the effects of which should be filtered out over time in any case). However, as well as modelling existing domain knowledge, the expert is attempting to add new classifications and justify them with a rule, or to find a new rule as a different way of making the same classification: this results in rules being added to the system which cannot be said to be correct or not, as the knowledge which is being represented by these rules is unconfirmed, and the accuracy of the knowledge is unknown to the expert. Therefore accuracy of the new rules is not a measure that can be used without extensive further research into the validity of each new rule. As the correctness for the classification of a case is determined by all of the classifications being provided, and due to the exception-based nature of MCRDR whereby newer rules can overrule previous rules, this makes determining the true classification accuracy of the system impossible once the knowledge base has been modified by a Knowledge Discovery process (rather than just Knowledge Acquisition). However the accuracy of the knowledge base for the Knowledge Acquisition phase can still be measured, although there are complexities involved.

Besides causing some oddities in the statistics when the expert is using the system for data mining (and therefore hypothesising that the classifications for a case may be incorrect or incomplete, even though they match the expert's current understanding of the domain), the classification accuracy of the system appears to follow a typical trend in MCRDR system development. Figure 3 shows that after an initial period of no correct classifications, the system steadily improves it's accuracy along a curve until it reaches 40 correct inferences. The departure at this point appears to reflect the first attempt of the expert to analyse the knowledge contained within the system, by comparing the classifications of different cases. After this the accuracy continues at a similar rate to before, although it stays at this rate rather than continuing to improve. The final stage, from approximately 75 correct inferences onwards, corresponded to when the expert changed focus to deriving new information from the dataset rather than classifying existing data.

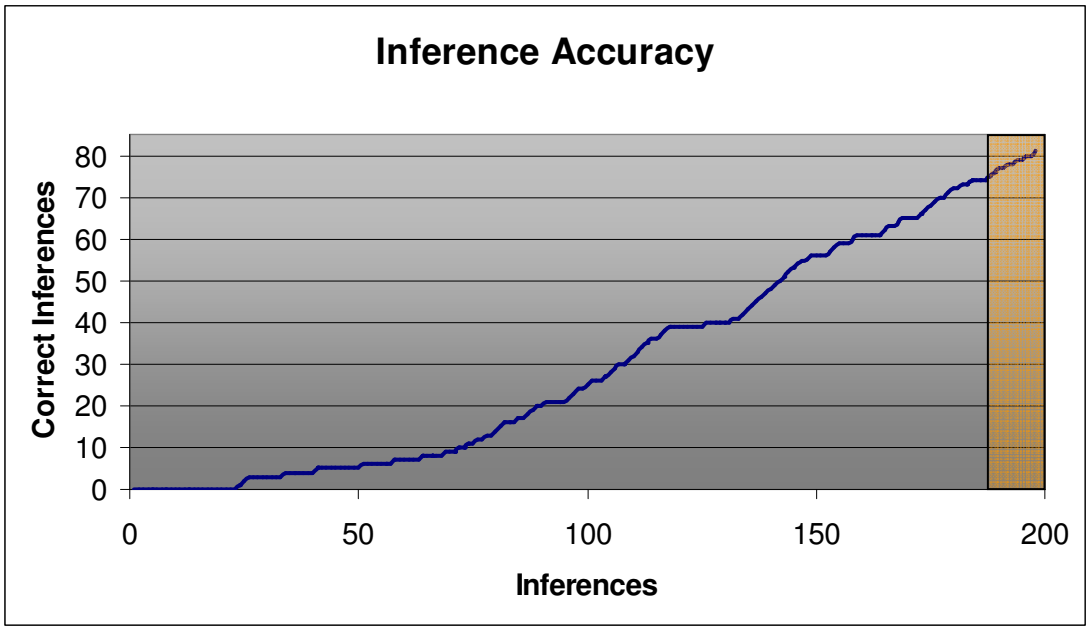

Figure 3: Inference Accuracy – Number of Correct Inferences against Total Number of Inferences Performed. The orange section shows when the expert was exploring for new knowledge

4.2.2. Rule Creation

The Knowledge Base grew steadily through use of the system to reach the level of 51 rules, with one section of rule deletion which was the result of a bug in the system, requiring the deletion and re-entry of a group of rules. Figure 5 shows a gradual increase in the rate of rule additions. This would appear to show that the expert becomes gradually accustomed to using the system. However, an examination of Figure 4 more correctly shows the learning curve for defining rules using the system: the time taken to define a rule is consistent at around 2 minutes per rule for the first 40 rules, after which the expert's familiarity with the system precludes a slow increase in efficiency over the next 10 (or so) rules when the expert can be said to be confident with rule creation and the rate of rule definition increases markedly. The slowing of the rule creation during the last phase corresponds to the expert beginning the data mining phase of trying to discover new rules: which is clearly, and predictably, a much more time consuming task than defining rules for known knowledge.

Instead, Figure 5 shows the gradual increase in understanding from the expert about the rules required to cover the domain: once the expert is satisfied that the rules existing in the knowledge base adequately classify the domain and dataset to allow for effective Knowledge Discovery, the rate of rule creation slows and ceases; the final tapering of the rate to the point of stopping completely shows the comparative length of time taken to derive new knowledge and form it into a rule.

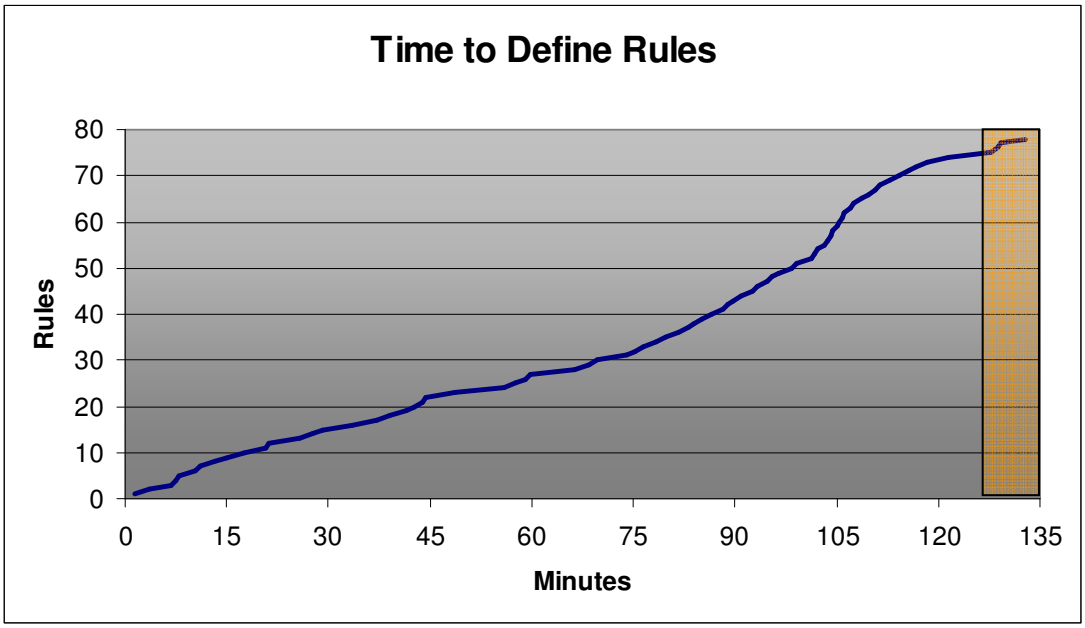

Figure 4: Time spent defining rules, with an average of 1.7 minutes per rule. The orange section corresponds to Knowledge Discovery rather than Acquisition

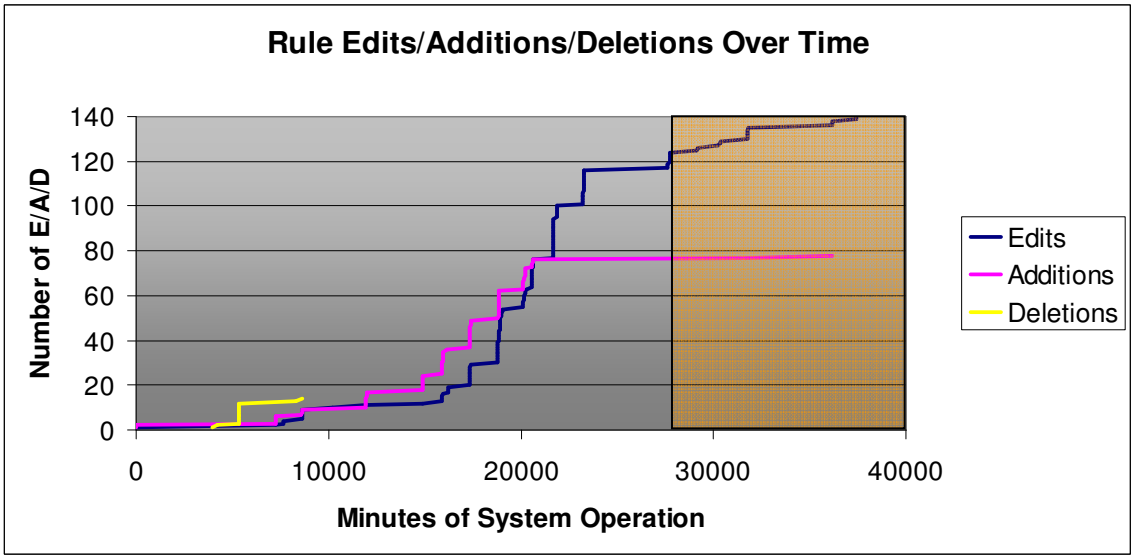

Figure 5: Numbers of Edits, Additions and Deletions throughout the use of the system, with the orange section showing when the expert changed from Knowledge Acquisition to Knowledge Discovery

4.2.3. Rule Edits

The editing of rules is one of the most significant results of this study, both because it is an unexplored area of the MCRDR method and because of the usage made of this functionality. Editing of rules was used extensively, 140 times in all, and at almost every stage of the process. The trend of the rule editing frequency Figure 5 is that the feature became more commonly used the further the system was in use. This continues until the point of beginning Knowledge Discovery phase, at which point the editing slows significantly, along with most other measures in the system. The figure also shows that in this phase, the expert almost exclusively uses rule editing to perform the Knowledge Discovery task, with only a very few rule additions. The reasons why rule editing is so frequent in this system, to the point of taking over from rule addition, is discussed later in Section 5.2.2 - Editing Rules.

A statistic for the time taken to edit a rule is unavailable as the data is confused with Knowledge Discovery attempts: as the Edit Rule functionality was the primary point for the expert to begin any examination of the dataset, which is a comparatively time consuming exercise, the data can not be said to be representative of actual rule editing times.

4.2.4. Rule Deletions

The deletions in the system are remarkable in that they are unremarkable: only 14 deletions occurred throughout the process, these occurring early on in the pure Knowledge Acquisition phase. These deletions were even then only in response to errors in the system which forced deletion and re-entry, and mistakes made by the expert through an incomplete understanding of how the Knowledge Acquisition process added to the knowledge base.

4.2.5. Rules per Conclusion

An interesting point of note in the knowledge base is that there is a 1:1 relationship between rules and conclusions – which is to say, no conclusion is used for more than one rule. This may have occurred in part because the expert can edit rules, and has therefore taken care to define complete rules for each classification that do not

exclude relevant cases. However, this does not account for situations where a conclusion can be based on two entirely different circumstances, that would require more than a single rule to define (it occurs in many domains that one classification can be reached from two very different groups of attributes (Compton & Edwards 1994; Richards, Chellen & Compton 1996) – this would require two rules as the system does not have any 'OR' logic implemented for rule conditions). The lack of conclusion reuse may instead be explained by the specificity of the conclusions themselves: in medical domains, any extra information that can be provided has the potential to change the final diagnosis (Bindoff 2005; Laszlo 1994), and therefore the expert has taken care to define conclusions which have as much meaning as possible. For example, the knowledge base contains conclusions which can be as specific (and similar) as "Reversibility (FEV1) and transfer factor >=80%" and "Reversibility (FEV1) and transfer factor <80%", which, while bearing similarities to the rule they are based from in some respects, provide a user with the detail required to make further decisions without having to take an extra step of examining the rule to discover why the classification was made.

4.3. Expert Review of the System

4.3.1. System Effectiveness

In their review of the system (see Appendix B – Expert Comments and System Review) the expert noted their observations on the comparative efficiency of using this method to using the previous method available to them, manual analysis via a commercial spreadsheet. The expert noted that to perform a relatively simple analysis previously required at least 3 hours, with much wasted effort due to errors made that required sections of the analysis to be repeated. Using the EMCRDR method the expert notes that the same analysis took 15 minutes to complete: a 12-fold increase in analysis efficiency. It was also noted that the MCRDR method was easier and far less prone to human error. The expert also made explicit note that the validation features of the system were particularly important in identifying unhelpful directions of study before significant time was wasted pursuing valueless avenues of research.

Also listed in the analysis of the system, the expert described the domain knowledge that was discovered in using the method. Three specific areas of study were examined, based on trends in the data available and the expert's understanding of the domain (for a fuller description of the exact Lung Function values and measurements being examined refer to Appendix A – System Usage Logs). In the first area being examined, the particular approach used by the expert to determine this involved creating rules with certain criteria to find the appropriate subsets of cases, then comparing these two sets. Upon finding that the criteria for one set gave a 60% probabilistic likelihood of the case appearing in the second set, the expert examined published literature on the subject and found studies confirming his results (D P Johns 2006/Appendix B – Expert Comments and System Review). While this means that the discovery is not totally new to the field, it is a significant result as it shows that an expert using this knowledge Discovery method can quickly find and provide evidence for hypotheses that would otherwise require substantial research and study. The expert then defined a rule combining the conditions of the previous two, and examined the similarities between those cases in other attributes. It was discovered that the inclusion of considering a certain other attribute increases the accuracy of using the simpler attribute to make the overall classification, although the expert notes that the work is still ongoing to attempt to make a classification certain. The relationship to this last attribute was discovered through the use of the 'Similarities' clustering tool provided by the system, showing an effective use of this method.

The second specific examination performed sought to explore an already known relationship between two groups – it found that one attribute (VA) provided a good estimate of another attribute (TLC) in cases without a certain classification (airflow obstruction), but consistently underestimated in cases with that classification. The relationship when plotted was found to be linear, and could be applied to estimate the value of one attribute (TLC) from another (VA). The relationship was again found to be similar to published data from an as yet unconfirmed study.

The third area of study sought to discover whether a known physiological effect (that obese people often have smaller lungs) could be consistently measured and classified using attributes in the dataset (by attempting to find whether the speed a person can blow out is a function of their Body Mass Index). The expert noted that preliminary

data suggests that there is no direct correlation between one classification (obesity) and the other (airflow obstruction), although there is such a correlation to another classification (small lungs).

4.3.2. Requested improvements

The expert expressed a desire to continue using the system in the future to discover more information, and had a number of suggestions as to how the system could be improved. Of first note is the request for further extensions to the clustering tool within the system: firstly the expansion of the information provided to include the mean and deviation of the subsets being examined, rather than just the range; and secondly the ability to define ranges explicitly to find relevant cases, rather than basing the ranges from selected cases.

The expert also made other suggestions which would have varying impacts on the method itself, such as the ability to combine rules. This could be handled as an interface feature, but highlights the focus of the expert on rule definition in Knowledge Discovery rather than case-based classification; and the ability to enter a rule without basing it explicitly on a case. Another improvement which was requested shows a difference made to the MCRDR process by the inclusion of a dataset: that the system provide feedback as to which cases in the dataset have not yet been classified. This shows that the normal case-based MCRDR Knowledge Acquisition is still of value to the expert, and perhaps highlights that the case-based classification approach cannot be abandoned entirely for a rule-based approach if the method is to maintain many of it's advantages.

Further improvements requested by the expert include interface improvements which are unrelated to the method itself, and so will not be discussed in detail in this study, but are noted in Appendix B – Expert Comments and System Review. A final suggestion made by the expert is that the system, with some modifications, would be particularly useful as a teaching and learning tool. This idea is discussed in Section 7 - Further Work.

5. Discussion

There has been definite success in data mining knowledge using the system: the three "new" rules (rules that the expert was unaware of) that were found show that the method can successfully discover knowledge which is both useful and otherwise difficult to discover. This section will include a discussion of the effectiveness and implications of the system.

5.1. Implications and Effectiveness of Modifications

5.1.1. Using a Dataset

Developing rules with the context of a complete dataset during the entire Knowledge Acquisition process, rather than presenting each case to the system individually to be classified by the expert, had a significant effect on the appearance and early efficiency of building the knowledge base. However, in terms of the expert's usage of the system and the development of the knowledge base, including the dataset made little difference. Basing all validation on every case in the dataset rather than only previously used cases (cornerstones) may have meant that the expert was more thorough in defining rules, as suggested by the higher than usual number of conditions per rule and frequent use of rule editing. While it certainly made performing validation slower for the first rules entered into the system – as a normal MCRDR validation process would not have any existing cornerstones with which to flag potential conflicts, making it much faster – the average time taken to define a rule was not significantly higher than any other MCRDR system. If the dataset is expanded to be significantly larger however, it may result that there will simply be too many cases to be able to validate against effectively. However, as there has been no sign that the validation was a problem with this implementation, this appears not to be a major enough concern to warrant changing the method at this stage.

Using the complete dataset for rule validation was of definite benefit when attempting to derive unconfirmed rules, as it allows the expert to view the effect of the rule in the context of the wider domain: without validation against the dataset (or a subset of the dataset) the expert could not have determined the relationships

between attributes that led to the new knowledge. If only a subset of the dataset were used (e.g. only cases stored because they were previously used as cornerstones) then not only would the dataset be irrelevant – simply a list of cases that are stored in preparation for being presented to the system – but the full effect of a new rule would not be apparent, particularly for statistical relationships. In normal MCRDR the list of cornerstones is used in a comparable way to the dataset in this system, in order to validate new rules, but this would not be effective for Knowledge Discovery.

5.1.2. Interface Modifications

5.1.2.1. Viewing the Knowledge Base

One of the most significant changes that was apparent in the use of the system was that the expert showed a tendency to want to consider what the impact of the rules they were adding would be to the knowledge base, rather than considering the impact of the rules on the output of the system, as is usually the case in MCRDR. That is to say that, when performing a Knowledge Acquisition process to build the knowledge base, the expert was focused on defining rules which represented his understanding of the domain rather than taking an individual case and defining rules so as to satisfy that the system classified this case correctly. This is evidenced by the manner in which the expert used the editing of rules in preference to rule creation, and the way the knowledge base consists of a single level of complex, considered rules.

There are a number of reasons why the expert may have focused on a rule-based Knowledge Acquisition rather than case-based. That the expert can and must view the knowledge base is suggested as a central cause: the expert is interested in what the domain knowledge looks like in a structured format. This means that the expert is likely also concerned about creating a knowledge base that is well-structured, appears logical, and is easy to understand. There may also be a level of mistrust that the program can correctly record what the expert is saying, or the expert may mistrust that they are using the system correctly, so there is a desire to check that what they have entered accurately represents how the domain works. This is probably an artefact of the usual MCRDR standard of presenting the user with an abstracted process of adding to the knowledge base: traditionally the rule creation is not represented in a format that makes it clear to the user what sort of an effect the rule will have on the knowledge base (including where the rule will be added). This

is usually appropriate in the case that the expert does not ever see the knowledge base. However if the user is expected to be able to understand the knowledge base, as is the case in this system, they will require a better understanding of how the knowledge base functions. The expert did express difficulty in understanding how to add a child rule in the correct place in the knowledge base (D P Johns 2006). Although he had an understanding of what the effect of a child rule would be, the expert did not understand how to add the rule as a child, in the desired place.

The normal, "intuitive" MCRDR format of hiding the knowledge base was maintained with this system as it was anticipated that it would be much easier for the expert to be able to build the knowledge base, as has been the case with other MCRDR Knowledge Acquisition implementations. However, due to the expert taking the structure of the knowledge base into greater consideration than was expected, the expert always attempts to define a rule which will fit their structured knowledge base, rather than adding rules via intuitive means. While this does have the advantage of allowing the expert to add to the knowledge base any rule, without having to wait for a relevant case to be presented to the system (which was one of the goals in creating a data mining tool), it changes the "intuitive" approach into an unintuitive approach, as the expert needs to understand how the knowledge base gets added to.

This may be a predilection of the expert that performed the testing, which will remain unknown unless different experts run through the Knowledge Acquisition process from the beginning. Whether this is true or not, it is an observed failing of the system that needs to be addressed. It may be possible that simply explaining the knowledge acquisition process in more detail will allow the expert to perform any desired task to either enhance the functionality of the system or produce a more structured knowledge base. However the complete lack of exceptions in the knowledge base, even once the expert had an understanding of the process, suggests that this would not be the case.

The other option for solving this problem is to change the interface for knowledge acquisition, by allowing the expert more freedom to explore that dataset, and to be able to freely define rules without being forced to have a specific case to base the

rule from. This would remove most of the content of the MCRDR Knowledge Acquisition approach, but does not imply that the approach be abandoned completely – the two approaches can be used together in one system to build the same knowledge base.

The problem presented by both potential solutions, and the essential difficulty that gave rise to the confusion, is that the expert is required to have a full understanding of the knowledge base structure and how it is extended – the expert will need to have most of the skills of a knowledge engineer, which may require extensive training – and negates one of the fundamental advantages of MCRDR Knowledge Acquisition and maintenance, that no knowledge engineer is required (Bindoff 2005; Compton & Edwards 1994). The problem may not be insurmountable, as providing the expert with training in how the knowledge base is structured and how to expand it is certainly an achievable task in many situations, particularly in using the system primarily as a data mining tool. However, if the system is also to be put into use as an expert system in the domain, implying use by multiple experts, or even if the data mining tool was to be a collaborative effort of Knowledge Acquisition, then the cost of training all experts in a good style of adding to the knowledge base could become prohibitive. This is particularly important as it is vital that all experts have a good understanding, and the same understanding, of what constitutes a well-structured knowledge base: if one expert adds rules according to a different view of how the knowledge base should be structured than the other experts, the knowledge base is likely to become confusing to view due to the context-focused approach to knowledge acquisition, which can result in a larger tree than otherwise if overly specific rules are entered in higher places in the tree (Kang & Compton 1992; Kang, Compton & Preston 1995). Another possibility to overcome this is to introduce a knowledge engineer into the process, who translates what the expert describes into rules, but this approach adds a layer of communication and point of failure, and the complication provided – particularly the time that would be required at each step – will make the system impractical for use in most real situations. The addition of a knowledge engineer would also significantly affect the maintainability of the system, both for use as a classification expert system and as a continuous data mining tool.

5.2. Implications and Effectiveness of Additional Features

5.2.1. Deleting Rules

The ability to delete rules is a further reinforcement to the need for the expert to understand the knowledge base structure. Usual MCRDR methods explicitly deny the ability of deleting rules from the knowledge base, as it can be complex to understand the full implications of removing a rule, particularly for an untrained user but even for a knowledge engineer: as MCRDR knowledge base nodes are entirely contextual, when removing any node other than a leaf node all the descendant rules from that point – and potentially the parent rule also – will be affected, and the greater the depth and branching of the sub-tree beneath that point, the more difficult it becomes to fully comprehend the effect that deleting the rule will have.

However, as previously noted, the expert in the building of this knowledge base did not make strong use of the rule deletion function. This is most likely due to the care with which the expert added rules to the system, ensuring that the knowledge base was well structured. The only deletions that took place were in a small number of cases to correct a misunderstanding of which interface controls caused a child to be added, but primarily in response to a bug which added rules as children incorrectly. In any other case when the expert made a mistake in building the knowledge base the rule edit ability was sufficient to correct the mistake, although it is noted that in theory it is possible to completely remove all the existing rules conditions and replace these, simulating a delete and add process, but this was not performed during testing. A point that was raised in discussing this with the expert was that, if the ability to move a rule from one place in the tree to another was provided, there would have been no use of the rule deletion function at all. The movement of rules will be discussed in the Further Work Section of this thesis. However, the lack of deletion may just be a factor of a limited time spent data mining with the system, as there is still potential benefit for removing rules which were thought to be interesting but were discovered not to be. While it is true that this form of rule deletion would only succeed in making the knowledge base cleaner (and the results returned cleaner, if the number of rules and classifications became too large to handle), this was clearly of great importance to the expert during this development. However, any rule

discovered could potentially be beneficial when combined with later discoveries, and so rule deletion should be avoided whenever possible. For these reasons, rule deletion is a feature that may still have potential use, but which should be discouraged unless the expert is certain that the rule has no benefit or is causing impossible confusion in the use and understanding of the system.

5.2.2. Editing Rules

Rule editing has similar requirements of the user to rule deletion, that is, that the user must have a good understanding of how the knowledge base functions. The expert will require a similar level of understanding to be able to edit rules as delete – in either circumstance at most one result will be affected, as only one branch of the knowledge base tree can be affected by changing one node or removing one node. Modifying a single rule can change a provided classification either by relaxing the rule's conditions (causing it to take the place of the parent's classification), or by changing the classification provided by the current rule by selecting a different classification from the list. More complex is the possibility that relaxing the conditions for a rule may cause any of its descendant rules to fire and provide their classification (which may be no classification, if it is a stopping rule), as they can now be accessed where previously they were not: this would replace the classification provided by the parent rule with potentially many classifications provided by the sub-tree of the rule being modified. Restricting the conditions for a rule may cause that rule's classification to not be applied to a case, and so the parent rule's classification will be applied instead. Understanding which of these scenarios will be the result of modifying a rule can be very difficult to calculate, particularly if the knowledge base is complex in form (has a large number of branches).

As has been noted, in the development of the knowledge base for this system the expert made extensive use of the rule edit feature to define and refine the knowledge base, and to search for new relationships. This is most likely also the cause of the shallowness of the knowledge base: as the expert wants to define a neater knowledge base rather than a potentially better functioning one, the expert will only enter rules that they understand. Further to this, when a rule is found to be incorrect the expert will then edit that rule rather than define an exception, as evidenced by the complete

lack of exception rules in the knowledge base, although the expert did find flaws in the knowledge base as the process was performed. Also, if rules are entered which do not make logical sense to the expert either in themselves or in structure, the expert will edit rules so that the knowledge base conforms to their understanding. The advantage of this is that the edit rule functionality provides a utility for experimenting with the knowledge base, and can therefore be used effectively by the expert to increase their understanding of how the knowledge base functions.

In the data mining phase of system use, the expert again used the edit rule feature extensively, in order to be able to view which cases were covered by the rule as it exists, and experiment with making small changes to the rule to see how this affected the case coverage from the dataset. It was also used as the primary access point for using the "similarities" data mining tool, as this tool requires the context of a particular rule to be of real benefit, for the same purpose. Although this was not predicted, the freedom of experimentation provided by this feature became a vital element to the data mining ability of the system. This is shown in that modifying an existing rule, or examining only the set of cases covered by a certain rule, were much more common actions when data mining than attempting to data mine with a new rule using the entire dataset.

This suggests that the system might benefit from the ability to transition directly from viewing a rule to performing data mining with the cases covered by that rule, without the intermediate step of the edit rule screen.

5.2.3. Data Mining Features

The specific data mining features of the system, in particular the 'Similarities' tool, proved very useful in the development of new hypotheses. The expert also found it particularly useful to be able to define a classification and export all those cases into a spreadsheet for further analysis.

However, the expert did note that the system would benefit from more complex and detailed automated analysis and statistics tools. This is to be expected: any further help which can be provided to the expert may be beneficial, and provided the tools

do not overload the expert with too much data to be able to efficiently examine, they should provide no detrimental effect. The expert also suggested that defining the ranges rather than selecting cases to generate the range would be of benefit. This shows a further inclination from the expert to move away from the case-based approach of normal MCRDR Knowledge Acquisition, to a more rule-based approach.

5.3. Impact on MCRDR Classification Ability

A potential impact of the changes to the MCRDR Knowledge Acquisition method that must be considered is that the accuracy of the knowledge base might be affected. This can not be determined in this system as the dataset has not been entirely verified, since one of the consequences of the expert taking a rule-based rather than case-based approach was that not every case was considered and confirmed to have the correct classifications. However, the classification accuracy statistics in Figure 3 show that the accuracy does increase, in a form similar to that of a regular MCRDR system as the Knowledge Acquisition progresses. The vital difference is that the accuracy does not taper towards the end, the usual sign that the knowledge base is approaching coverage of the domain. This suggests that either the accuracy of the method has been affected, or that the knowledge base is simply incomplete. The uncomplicated nature of the knowledge base would seem to support the latter idea, as a complete knowledge base would be expected to be more complex in structure. It can also be noted that even if the accuracy of the method for providing an expert system is compromised by the modifications made, or if the knowledge base is incomplete, it would still appear to be sufficient for use as a basis for some degree of Knowledge Discovery. However this is only conjecture until the knowledge base can be verified for the domain or dataset.

6. Conclusions

The preliminary tests performed using the prototype system developed suggest that the MCRDR method, with substantial modification, can be applied successfully to Knowledge Discovery tasks. This new method of Exposed MCRDR allowed an expert in the domain of Lung Function to discover and provide some measure of evidence for knowledge that the expert was not previously aware of, in a much shorter time than the manual process which would previously have been performed. However no comparison has been made to other Knowledge Discovery methods – while many other methods are not applicable to the types of domain which EMCRDR was designed for, until a comparison is made the real benefit of the method to the field of Knowledge Discovery cannot be accurately measured, although the simple fact that this kind of tool is rare for domains such as the one considered suggests that this study may be of significant value.

Although the method has been shown to work, there are a number of improvements that may enhance the applicability and effectiveness of the method for Knowledge Discovery purposes. In particular the assistance provided to the expert in discovering trends in the data is an area that can always be diversified and improved, until the point at which the extra information about the data provided becomes too large for the expert to be able to easily comprehend. The confusion of the expert in using the case-based approach to Knowledge Acquisition also denotes another area that can be improved: allowing the expert more freedom in defining rules may alleviate the problem. The other option is to provide a more detailed and thorough explanation to the expert of how the process functions, although results suggest that this would not resolve the issue.

One of the issues raised by this system was that the method does require that the expert have a level of knowledge about how the process works beyond that which is normally required by an MCRDR Knowledge Acquisition method. However, the proficiency of the expert was shown to increase to a sufficient level in a very short space of time, so that the expert was able to make effective use of the system without anything more than basic instruction and experimentation. It can also be argued that

any other Knowledge Discovery method would also require a familiarisation period at least equal to that of this method, but this is unproven conjecture.

Perhaps the most significant result to be attained from this study is not that a successful knowledge base was produced using this method, including new knowledge, but rather the process by which that knowledge base was constructed, and what the implications are for MCRDR Knowledge Acquisition. The extra functionality allowed to the expert in developing the knowledge base had a dramatic effect on how the Knowledge Acquisition was performed: allowing the expert to edit rules, combined with viewing the dataset, fundamentally altered the manner in which the expert approached the Knowledge Acquisition from a case-based perspective to a rule-based perspective. This would confirm previous literature that suggests that allowing rule editing would not be of benefit in Knowledge Acquisition for expert system development. However if the goal is to develop a knowledge base that is readable and useful to a human, these features show a definite positive move towards this goal. A possible disadvantage is that they may detract from the accuracy of the system. Allowing rule deletions had little impact on this study, but a more thorough test might provide a better analysis of the usefulness of this feature.

The benefits of the method modifications are in some ways less clear than the disadvantages. While showing the knowledge base influences the method of its construction, it allows the expert to review the recorded knowledge to find missing knowledge, find areas to explore, and review flaws. Similarly while rule editing changes the perception of how to create the knowledge base, it allows the knowledge base to be expressed in a form which the expert can easily understand, enhancing (or possibly making plausible) the advantage of showing the knowledge base. It is theorised that allowing rule deletions would further enhance this ability, but the inherent risks in damaging the knowledge may outweigh the benefits. Further analysis needs to be performed to test this.

In final conclusion, despite remaining uncertainties about how the method should be implemented, it can be seen that the Exposed MCRDR method is a valuable Knowledge Discovery approach, even for domains which require extensive background knowledge, have large volumes of difficult to interpret cases, and have

complex and specific target knowledge. In the short time that an expert was using the method to model the domain and discover new data, three useful and previously unknown methods of classification were derived. However, determining the full potential of the method, and how to achieve that potential, requires significant further testing. The insights provided by this method into the MCRDR Knowledge Acquisition process also require further examination to determine their extent and applicability. The following section will discuss these future directions for the method and MCRDR Knowledge Acquisition.

7. Further Work

The discoveries made in this study lead to many areas with the potential for further research. The success of the algorithm as a data mining assistance application require further verification and analysis before a conclusive statement can be made about its overall usefulness. However, the discoveries about how modifications to the MCRDR algorithm affect its use are of particular interest. The modifications and their potential applications are an unexplored area which has shown unexpected and interesting results.

7.1. Further EMCRDR Evaluation

The EMCRDR method has been shown to be an effective tool for assisting in the discovery of previously unknown knowledge in the particular prototype system that was developed. However, the overall usefulness is difficult to determine. How the data mining ability will extend beyond the first few examinations made by the expert is unknown, and very difficult to estimate. Therefore the method needs to be examined over different or larger datasets before a declarative statement can be made. This would possibly also reveal more information about the effectiveness of the modifications made to the MCRDR process such as rule deletion, which may have been under-utilised simply due to a lack of opportunity from the size of the dataset.

It is of particular importance to note that, while it has been measured to the best extent possible, the accuracy of the knowledge base (excluding the newly discovered knowledge, as this can not be measured except by studies with more source data for evidence) can not be fully determined without verifying that the system returns correct and complete classifications for all cases in the dataset. As has been previously noted this does not necessarily affect the discovery of new knowledge, which only requires sufficient background knowledge to be able to support the hypotheses. However, to fully show how the modifications made to the MCRDR process affect the Knowledge Acquisition, and to determine how useful the produced knowledge base can be, the full dataset verification needs to be performed. This work is relatively simple, if potentially time consuming. An expert is needed to examine

each case in the set individually, and test to see that the system provides correct classifications for every case. In the situation that a case is classified incorrectly, or incompletely, the expert must justify why this is so in the form of a new rule which is added to the knowledge base. Once this is completed the knowledge base can be said to entirely cover the domain as represented by that dataset, and an analysis of how effective the method is at modelling the domain and at classifying cases can be presented.

Another vital measurement that has not been performed is the effectiveness of the method as compared to other data mining methods. In particular a comparison to clustering techniques should be performed. Such clustering techniques are suggested as the only Knowledge Discovery methods which are likely to be capable of functioning effectively within a domain such as Lung Function which contains unclassified and detailed cases, and complex target relationships. A comparative analysis is required to be able to state with any conviction whether this method is a useful approach to Knowledge Discovery, rather than just an effective method.

7.2. EMCRDR Enhancements

There are many enhancements that can yet be made to improve the effectiveness and ease of use of the system. The most immediately useful are extensions to the Machine Learning tools and abilities present in the software, to allow the expert to more easily find trends in the data which may indicate a potential means of justifying a hypothesis or indicate an entirely new relationship. The expert has requested that the mean and standard deviation for each attribute in the subset be added to the existing similarity finding screen, and that the tool allow the explicit definition of ranges (Appendix B – Expert Comments and System Review).

A further enhancement that might be able to be made in this area is the automatic generation of rules using a standard clustering algorithm. It might be possible that using the classifications provided by an expert, possibly in combination, will restrict the dataset enough that automated clustering techniques can find rules from patterns in the data. The problem remains that if too many rules are generated, with many complex and irrelevant rules, then the expert will not be able to efficiently sort

through and determine which are useful and which are not. However the direction is one which can definitely be explored, as the potential benefits to the speed of the process are high.

This also raises the possibility of separating the process into two distinct phases: first a Knowledge Base Building phase; and secondly a Data Mining phase. The first phase would consist of normal MCRDR Knowledge Acquisition to build the knowledge base with appropriate domain knowledge, while the second phase would make use of the EMCRDR method, or possibly even another method which could make use of the MCRDR knowledge base to derive new knowledge. The difficulty presented by this approach would be in the analysis of the knowledge base produced, as knowledge bases produced by regular MCRDR Knowledge Acquisition can be counter-intuitive and difficult to understand. Determining at which point the knowledge base has reached a sufficient level of knowledge is also difficult. However this approach can potentially provide a number of benefits. Firstly, breaking the process into two phases would ensure that a strong knowledge base was built initially before any Knowledge Discovery was performed, increasing the probability that new knowledge will be found and providing a measure of assurance that any knowledge discovered is valid. It also reduces the concern that the expert will affect the accuracy of the knowledge base during the Knowledge Discovery process.

A consideration that is presented by this is that the converse approach could be useable – that is, using EMCRDR to build a knowledge base and then expanding this knowledge base using regular MCRDR. This should produce an effective knowledge base, possibly with the benefit of having an understandable top level, although this is usually true in any MCRDR knowledge base. Besides this, it may be a useful study to assist in determining how effective the knowledge base created by the EMCRDR method is, by analysing how many rules are added via the regular MCRDR Knowledge Acquisition method (i.e. rules that were missed by the EMCRDR phase).

The expert also requested that the system display the cases which have not yet been classified by the rules in the system – this could be used as a basic tool for finding extreme cases that elude classification, but as each case generally receives multiple

classifications this feature would only be a partial measure of ensuring completeness compared to inferencing and classifying each case individually. This feature would be unnecessary in any case were either of the above two suggestions taken into consideration, as the dedicated MCRDR Knowledge Acquisition process would outperform this measure in achieving the desired goals.

There are also many simpler but potentially much more effective improvements that can be made to the applicability of the method. Expanding the capability of the database and system to be able to handle time series data is an area of particular use in this domain and most other medical domains, to be able to analyse the deterioration or improvement of a patient's condition over time, and possibly in conjunction with the treatment provided (although this system does not incorporate treatment recommendations at this stage).

Another simple enhancement to the implementation of the system, which can have a dramatic effect on both the scope and benefits of the results provided, is the ability to handle missing values in the dataset. This is a complication that is present in many real datasets, particularly in the field of Lung Function where full tests are rarely performed, and so the vast majority of the data stored is incomplete. Improving the system to be able to handle this is a necessary improvement if the method is to be able to fully analyse the Lung Function domain or similar domains.

7.3. MCRDR Modifications and Additions

Besides further work into the EMCRDR Knowledge Discovery method, this study has also revealed potential modifications to the regular MCRDR Knowledge Acquisition method. The effect of being able to edit rules is difficult to describe effectively without further work into the area, particularly without a more complete analysis of how accurate the produced knowledge base is over the full dataset. However the frequency of use of the feature made by the expert alone suggests that this feature may be a worthwhile consideration for MCRDR implementations. Despite potential benefits, further work must be performed to analyse the full effect of the edit rule feature before it can be said that it would explicitly increase or decrease the effectiveness of the method as a whole, although it is acknowledged that

adding such a feature to a traditional MCRDR system may have complications on validation which might detract from the long-term maintainability of the system. The rule deletion results are similarly inconclusive at this stage: there were no apparent negative effects of allowing this feature, but the feature was not actively used. The potential for use of this feature requires further examination.

A common feature of the how the expert described their analysis of the Lung Function domain was that they would consider making new classifications in terms of other classifications – a simplistic example being creating a rule to define obesity based on the Body Mass Index attribute, and then creating a new rule which (in the expert's speech) used "obesity" as one of the conditions. This is an illustration of the nature of the specific domain as a *classification domain* rather than a *diagnostic domain*. In such a classification domain no classification is necessarily an end result – it is only an indicator of certain characteristics of a case. This use of classifications as conditions could be implemented in two ways: performing multiple inferences, either on a single knowledge base tree or using a hierarchy of knowledge base trees; or by allowing the definition of rules not only as exception rules but optionally as "additional" rules, which add their classification to the case without excluding the parent classification. This second method was explicitly suggested by the expert in a discussion towards the end of system operation (demonstrating that the expert had a strong level of understanding of how the knowledge base functioned and an interest in how it might be improved). As a further extension to this, the complexities of conjoining multiple classifications as conditions in one rule would make an interesting study that may provide the MCRDR algorithm with a wider range of applicability.

7.4. Educational Applications

This system is perhaps best described as a learning/education tool: the system allows a user to define their knowledge in the form of a knowledge base, and to then explore how they can build upon that knowledge base by discovering new information about the domain. When the user is an expert in the field it can be assumed that new information will be of benefit not only to the expert but also to others in the field, however, the potential of the system simply as an educational learning tool for any

user should also be considered. This potential is further enhanced when an expert is involved first to develop the knowledge base, then the system provided to an inexperienced student: the student can explore the differences in the responses the system provides to their own evaluation of cases, and the system can provide a justification of why the differences exist. If the system were to be adapted specifically for an educational role, there are many enhancements that might be included, such as a more user-friendly justification of why each classification was made, including why some classifications were not made in the case of relevant stopping rules, and who entered the rule (e.g. teacher or student) providing a measure of reliability to the result.

The system, or potentially any MCRDR system, could also be used educationally as a common repository of classification knowledge and strategy between multiple experts. This would be beneficial in ensuring all expertise is shared by all experts, in standardizing the method of classification between experts, and in determining (and resolving, if possible) conflicts in opinion.

7.5. Summary

This study has revealed potential for extensive further research, both into the improvement, further development and applicability of the EMCRDR Knowledge Discovery method, but also into the enhancement of the MCRDR Knowledge Acquisition method. If the EMCRDR method can be verified, proven and enhanced where required, it may provide a solution to performing Knowledge Discovery in complex domains which would otherwise be a very slow and difficult process. The verification process performed on the EMCRDR method should also provide more complete data on the implications of the modifications made to the MCRDR Knowledge Acquisition process, allowing conclusions to be made about when and how those modifications might be useful in MCRDR system development.

8. References

Aamodt, A & Plaza, E 1994, 'Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches', *AI Communications*, vol. 7, no. 1, pp. 33-59.

Aaron, S, Dales, R & Cardinal, P 1999, *How Accurate Is Spirometry at Predicting Restrictive Pulmonary Impairment?**.

Abe, H & Yamaguchi, T 2005, 'Implementing an Integrated Time-Series Data Mining Environment - A Case Study of Medical KDD on Chronic Hepatitis', paper presented to First International Conference on Complex Medical Engineering (CME2005), Takamatsu, Kagawa, Japan.

Aha, DW 1991, 'Case-Based Learning Algorithms', paper presented to Proceedings of the DARPA Case-Based Reasoning Workshop, Washington D.C.

Bachant, J & McDermott, J 1984, 'RI Revisited: Four Years in the Trenches', *AI Magazine*, vol. 5, no. 3, pp. 21-32.

Barker, V, O'Connor, D, Bachant, J & Soloway, E 1989, 'Expert systems for configuration at Digital: XCON and beyond', *Communications of the ACM*, vol. 32, no. 3, pp. 298-318.

Bindoff, IK 2005, 'An Intelligent Decision Support System for Automated Medication Review', University of Tasmania.

Buchanan, B & Shortliffe, E 1984, *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*, Addison-Wesley.

Buchanan, B, Barstow, D, Bechtal, R, Bennett, J, Clancey, W, Kulikowski, C, Mitchell, T & Waterman, D 1983, 'Constructing an expert system', *Building Expert Systems*.

Catlett, J 1992, 'Ripple down rules as a mediating representation in interactive induction', paper presented to Proceedings of the Second Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, Kobe, Japan, Nov 9-13.

Chi, RH & Kiang, MY 1991, 'An integrated approach of rule-based and case-based reasoning for decision support', in *Proceedings of the 19th annual conference on Computer Science*, ACM Press, San Antonio, Texas, United States, pp. 255-67.

Clancey, WJ 1984, 'Knowledge acquisition for classification expert systems', in *Proceedings of the 1984 annual conference of the ACM on The fifth generation challenge*, ACM Press, pp. 11-4.

---- 1993, 'Situated action: A neuropsychological interpretation (Response to Vera and Simon)', *Cognitive Science*, vol. 17, no. 1, pp. 87-116.

Clerkin, P, Cunningham, P & Hayes, C 2001, 'Ontology Discovery for the Semantic Web Using Hierarchical Clustering', *Semantic Web Mining Workshop at ECML/PKDD-2001, September*, vol. 3.

Compton, P & Jansen, R 1989, 'A philosophical basis for knowledge acquisition', paper presented to European Knowledge Acquisition for Knowledge-Based Systems, Paris.

Compton, P & Edwards, G 1994, 'A 2000 Rule Expert System Without a Knowledge Engineer', paper presented to Proceedings of the 8th AAAI-Sponsored Ban Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada.

Compton, P, Cao, T & Kerr, J 2004, 'Generalising Incremental Knowledge Acquisition', paper presented to Pacific Rim Knowledge Acquisition Workshop, Auckland, 9th - 10th August.

Compton, P, Kang, B, Preston, P & Mulholland, M 1993, 'Knowledge Acquisition without Analysis', paper presented to Knowledge Acquisition for Knowledge-Based Systems, Springer Verlag.

Compton, P, Edwards, G, Kang, B, Lazarus, L, Malor, R, Preston, P & Srinivasan, A 1992, 'Ripple down rules: Turning knowledge acquisition into knowledge maintenance.' *Artificial Intelligence in Medicine*, vol. 4, no. 6, pp. 463-75.

D P Johns, P, FANZSRS 2006, to T Ling, July 2006.

Dazeley, R & Kang, B 2004, 'An Online Classification and Prediction Hybrid System for Knowledge Discovery in Databases', paper presented to The 2nd International Conference on Artificial Intelligence in Science and Technology, Hobart, Australia.

Edwards, G, Compton, P, Malor, R, Srinivasan, A & Lazarus, L 1993, 'PEIRS: a pathologist-maintained expert system for the interpretation of chemical pathology reports', *Pathology*, no. 25, pp. 27-34.
Feigenbaum, EA 1977, *The art of artificial intelligence: I. Themes and case studies of knowledge engineering*, Stanford University.
Féret, M & Glasgow, J 1997, 'Combining Case-Based and Model-Based Reasoning for the Diagnosis of Complex Devices', *Applied Intelligence*, vol. 7, no. 1, pp. 57-78.
Ferguson, G, Enright, P, Buist, A & Higgins, M 2000, *Office Spirometry for Lung Health Assessment in Adults* A Consensus Statement From the National Lung Health Education Program*.
Frawley, W, Piatetsky-Shapiro, G & Matheus, C 1992, 'Knowledge Discovery in Databases: An Overview', *AI Magazine*, vol. 13, no. 3, pp. 57-70.
Gaines, B & Boose, J 1988, *Knowledge Acquisition for Knowledge-Based Systems*, vol. 1, Academic Press, Inc., Orlando, FL, USA.
Gaines, B & Compton, P 1992, 'Induction of Ripple Down Rules', paper presented to Proceedings of the 5th Australian Conference on Artical Intelligence, Hobart, Australia.
Gaines, BR 1993, 'Modeling and Extending Expertise', in *Proceedings of the 7th European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Springer-Verlag, pp. 1-22.
Gerber, A, Glynn, E, MacDonald, A, Lawley, M & Raymond, K 2004, 'Modelling for Knowledge Discovery', paper presented to Workshop on Model-driven Evolution of Legacy Systems (MELS).
Glady, C, Aaron, S, Lunau, M, Clinch, J & Dales, R 2003, *A Spirometry-Based Algorithm To Direct Lung Function Testing in the Pulmonary Function Laboratory**.
Goebel, M & Gruenwald, L 1999, 'A survey of data mining and knowledge discovery software tools', *SIGKDD Explor. Newsl.*, vol. 1, no. 1, pp. 20-33.
Goldberg, DE & Holland, JH 1988, 'Genetic Algorithms and Machine Learning', *Machine Learning*, vol. V3, no. 2, pp. 95-9.
Golding, A & Rosenbloom, P 1996, 'Improving accuracy by combining rule-based and case-based reasoning', *Artificial Intelligence*, vol. 87, no. 1, pp. 215-54.
Grefenstette, J, Ramsey, C & Schultz, A 1990, 'Learning sequential decision rules using simulation models and competition', *Machine Learning*, vol. 5, no. 4, pp. 355-81.
Hall, M & Smith, L 1998, 'Practical feature subset selection for machine learning', paper presented to Proceedings of the 21st Australasian Computer Science Conference.
Hand, D & Vinciotti, V 2003, 'Choosing k for two-class nearest neighbour classifiers with unbalanced classes', *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1555-62.
Hong, T-P, Wang, T-T, Wang, S-L & Chien, B-C 2000, 'Learning a coverage set of maximally general fuzzy rules by rough sets', *Expert Systems with Applications*, vol. 19, no. 2, pp. 97-103.
Hughes, D & Empey, D 1981, *Lung Function for the Clinician*, Academic Press Grune and Stratton, London.
Ihrig, L & Kambhampati, S 1995, 'An Explanation-Based Approach to Improve Retrieval in Case-Based Planning', paper presented to European Workshop on Planning (EWSP-95).
Kang, B & Compton, P 1992, 'Knowledge Acquisition in Context: the Multiple Classification Problem', paper presented to Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Seoul.
Kang, B, Compton, P & Preston, P 1995, 'Multiple Classification Ripple Down Rules: Evaluation and Possibilities', paper presented to Proceedings 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, Banff, Feb 26 - March 3.
---- 1998, 'Simulated Expert Evaluation of Multiple Classification Ripple Down Rules', paper presented to 11th Banff knowledge acqusition for knowledge-based systems workshop, University of Calgary.
Kang, B, Yoshida, K, Motoda, H & Compton, P 1997, 'Help desk system with intelligent interface', *Applied Artificial Intelligence*, vol. 11, no. 7-8, pp. 611-31.
Kolodner, J 1991, 'Improving Human Decision Making through Case-Based Decision Aiding', *AI Magazine*, vol. 12, no. 2, pp. 52-68.

Kowalski, A 1991, 'Case-based reasoning and the deep structure approach to knowledge representation', in *Proceedings of the 3rd international conference on Artificial intelligence and law*, ACM Press, Oxford, England, pp. 21-30.
Kurniawati, R, Jin, J & Shepherd, J 1998, 'Efficient nearest-neighbour searches using weighted euclidean metrics', *Proceedings of the 16th British National Conferenc on Databases: Advances in Databases*, pp. 64-76.
Laszlo, G 1994, *Pulmonary FunctionA Guide for Clinicians*, Cambridge University Press, Cambridge.
Leondes, C 2002, *Expert Systems*, vol. 4, 6 vols., Academic Press, San Diego.
Liou, YI 1990, 'Knowledge acquisition: issues, techniques, and methodology', in *Proceedings of the 1990 ACM SIGBDP conference on Trends and directions in expert systems*, ACM Press, Orlando, Florida, United States, pp. 212-36.
Manago, M, Althoff, K, Auriol, E, Traphoner, R, Stefan, W, Conruyt, N & Maurer, F 1993, 'Induction and Reasoning from Cases', paper presented to First European Workshop on Case-Based Reasoning, Kaiserslautern, Germany.
MedGraphics Pulmonary Consult™ Software, 2006, Medical Graphics Corporation, viewed 10/13/2006 2006, <http://www.medgraphics.com/datasheet_pconsult.html>.
Miller, A 1987, *Pulmonary Function Tests*, Grune and Stratton, Orlando.
Mitchell, T 1997, 'Artificial Neural Networks', in T Mitchell (ed.), *Machine Learning*, McGraw-Hill, pp. 82-112.
Oswald, H, Phelan, P, Lanigan, A, Hibbert, M, Carlin, J, Bowes, G & Olinsky, A 1997, 'Childhood asthma and lung function in mid-adult life.' *Pediatr Pulmonol*, vol. 23, no. 1, pp. 14-20.
Paris, C & Gil, Y 1993, 'EXPECT: Intelligent Support for Knowledge Base Refinement', in *Proceedings of the 7th European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Springer-Verlag, pp. 220-36.
Pribor, H 1989, 'Expert systems in laboratory medicine: A practical consultative application', *Journal of Medical Systems*, vol. 13, no. 2, pp. 103-9.
Punjabi, N 1998, 'Correction of single-breath helium lung volumes in patients with airflow obstruction', *Chest*, vol. 114, no. 3, pp. 907-18.
Quinlan, J 1986, 'Induction of decision trees', *Machine Learning*, vol. 1, no. 1, pp. 81-106.
---- 1993, *C4.5: Programs for Machine Learning*, vol. 1, 1 vols., Morgan Kaufmann Publishers Inc., Sydney.
---- 1996, 'Bagging, boosting, and C4. 5', paper presented to Proceedings of the Thirteenth National Conference on Artificial Intelligence.
Quinlan, JR 1987, 'Simplifying decision trees', *Int. J. Man-Mach. Stud.*, vol. 27, no. 3, pp. 221-34.
Richards, D & Compton, P 1997, 'Combining Formal Concept Analysis and Ripple Down Rules to Support the Reuse of Knowledge', paper presented to Proceedings Software Engineering Knowledge Engineering SEKE'97, Madrid, June 18-20.
Richards, D & Busch, P 2003, 'Acquiring and Applying Contextualised Tacit Knowledge', *JOURNAL OF INFORMATION AND KNOWLEDGE MANAGEMENT*, vol. 2, pp. 179-90.
Richards, D, Chellen, V & Compton, P 1996, 'The Reuse of Ripple Down Rule Knowledge Bases: Using Machine Learning to Remove Repetition', paper presented to Proceedings of Pacific Knowledge Acquisition Workshop PKAW'96.
Roberto J. Bayardo, J & Agrawal, R 1999, 'Mining the most interesting rules', in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, San Diego, California, United States, pp. 145-54.
Ruppel, GL 1994, *Manual of Pulmonary Function Testing*, 7 edn, Mosby, St Louis.
Sester, M 2000, 'Knowledge acquisition for the automatic interpretation of spatial data', *International Journal of Geographical Information Science*, vol. 14, no. 1, pp. 1-24.
Shaheen, S, Sterne, J, Tucker, J & Florey, C 1998, *Birth weight, childhood lower respiratory tract infection, and adult lung function*, British Thoracic Society.
Singh, PK 2006, 'Knowledge-based Annotation of Medical Images', The University of New South Wales.
Snow, M, Fallat, R, Tyler, W & Hsu, S 1988, 'Pulmonary Consult: Concept to application of an expert system', *Journal of Clinical Engineering*, vol. 13, no. 3.

Srinivasan, A, Compton, P, Malor, R, Edwards, G & Lazarus, L 1992, 'Knowledge Acquisition in Context for a Complex Domain', paper presented to Proceedings of the Fifth European Knowledge Acquisition Workshop.
Swanney, M, Beckert, L, Frampton, C, Wallace, L, Jensen, R & Crapo, R 2004, *Validity of the American Thoracic Society and Other Spirometric Algorithms Using FVC and Forced Expiratory Volume at 6 s for Predicting a Reduced Total Lung Capacity**.
Towell, G & Shavlik, J 1994, 'Knowledge-based artificial neural networks', *Artificial Intelligence*, vol. 70, no. 1, pp. 119-65.
Towell, GG & Shavlik, JW 1993, 'Extracting refined rules from knowledge-based neural networks', *Machine Learning*, vol. V13, no. 1, pp. 71-101.
Tsumoto, S 1998, 'Modelling medical diagnostic rules based on rough sets', paper presented to Proceedings of the First International Conference on Rough Sets and Current Trends in Computing.
---- 2000, 'Automated discovery of positive and negative knowledge in clinical databases', *Engineering in Medicine and Biology Magazine, IEEE*, vol. 19, no. 4, pp. 56-62.
---- 2004, 'Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model', *Information Sciences*, vol. 162, no. 2, pp. 65-80.
Watson, I & Marir, F 1994, 'Case-Based Reasoning: A Review', *The Knowledge Engineering Review*, vol. 9, no. 4, pp. 327-54.
Witten, IH & Frank, E 2000, *Data Mining: Practical Machine Learning Tools with Java Implementations*, Morgan Kaufmann Publishers, San Diego.
Yamaguti, T & Kurematsu, M 1993, 'Legal knowledge acquisition using case-based reasoning and model inference', in *Proceedings of the 4th international conference on Artificial intelligence and law*, ACM Press, Amsterdam, The Netherlands, pp. 212-7.

9. Appendix A – System Usage Logs

The following is a description of each of the logs recorded by the system:

PERFORM_INFERENCE: Records the action of the user asks the system to provide the classifications for a case, with the caseID of the case being inferenced, and the number of conclusions returned.

ACCEPT_CONCLUSIONS: Records the user taking no further action (returning to the dataset) after they have performed an inference. The implication is that the expert has viewed the conclusions provided for the case, and determined that there are no errors or missing conclusions. The caseID is stored with this record.

ADD_CONCLUSION: Records the action of the expert deciding to add a conclusion for a case, when they have determined that the system's knowledge is incomplete for this case. The practical result of deciding to add a conclusion is to be taken to the screen for adding a rule, based on the current case. This record stores the caseID for the case being used as the cornerstone, and the conclusionID for the conclusion being added (i.e. the conclusion of the new rule).

CHANGE_INCORRECT_CONCLUSION: Records the action of the expert determining that a conclusion provided by the system for a certain case is incorrect, and needs to be changed. Practically this involves the expert adding a new rule as a child to the rule(s) that caused the incorrect conclusion to be given. The log record contains the caseID, conclusionID of the new conclusion being added, and the ruleIDs of all rules that caused the incorrect conclusion.

REMOVE_INCORRECT_CONCLUSION: Records when the expert inferences a case, and decides that one of the conclusions is incorrect, and should not be displayed for this case. The practical result of this is that the expert adds a stopping rule as a child of the rule(s) that caused the incorrect conclusion to be made. The log record contains the caseID and ruleID(s).

DEFINE_NEW_CONCLUSION: Records the action of the expert defining a new conclusion in the system, either as part of the process of adding a rule or just adding it independently for future use. Stores the new conclusionID.

DELETE_CONCLUSION: Records the deletion of a conclusion from the system, independent of any rule. Stores the conclusionID of the deleted conclusion.

EDIT_CONCLUSION: Records the expert modifying the title or description of a conclusion (independent of any rule that the conclusion is used in). Stores the conclusionID of the conclusion being edited.

DEFINE_CLAUSE: Records the definition of a new condition for a rule, with the attribute used, the operator, and the value. This is performed only when the rule is validated or saved, so as to avoid logging input errors and only store conditions that the expert considers meaningful enough to perform a validation upon.

EDIT_CLAUSE: Records when a condition for a rule is edited, with the old attribute, operator, and value, and the new attribute, operator and value. This is performed only when the rule is validated or saved, so as to avoid logging input errors and only store conditions that the expert considers meaningful enough to perform a validation upon.

DELETE_CLAUSE: Records the deletion of a condition from a rule. The attribute, operator and value are all stored.

VALIDATE_RULE: Records whenever the system performs validation upon a rule. This includes automatic validation performed by the system when the rule is displayed, and also validation performed explicitly by the user in order to test rule modifications. The log record stores the number of clauses in the rule at the time of validation, and the number of cases that the rule currently covers (the results of the validation being performed).

SAVE_RULE: Records the action of saving a rule in its current state, with the ruleID to identify it.

CANCEL_CREATING_RULE: Recorded when the expert leaves the rule definition screen via the cancel button, when defining a new rule.

CANCEL_EDITING_RULE: Records the action of leaving the rule definition screen via the cancel button, when editing an existing rule. The log record stores the ruleID of the rule that was being edited.

EDIT_RULE: Records the action of choosing to edit a rule, with the ruleID of the rule being edited.

DELETE_RULE: Records the action of the expert deleting a rule from the knowledge base, including the ruleID and whether the expert chose to delete the child rules or not.

QUERY_SIMILARITIES: Records when the user chooses to go to the similarities screen, and stores the caseIDs of all the cases that currently match the rule (the cases that will be examined for similarities).

10. Appendix B – Expert Comments and System Review

Comments

Overview:

There is an enormous amount of information stored in current and archived databases held within respiratory laboratories throughout the world. The majority of this data is recorded numerically and was obtained during routine clinical testing of people referred for assessment, usually because they were suspected of having lung disease or had a confirmed diagnosis.

To a very large extent the data contained in these databases remain 'dormant' despite the fact that they represent an enormous clinical, scientific and, in particular, teaching resource.

In broad terms, the aim of this Honours project was to develop a 'proof of concept' database model (MCRDR) to accept lung function test results, and provide an engine for 'intelligent' physiological interpretation (classification) of lung function test data, including a user interface to enable the test data to be easily interrogated to facilitate the acquisition of new knowledge.

Specific Comments on the Project:

A custom MCRDR engine to analyse lung function data was successfully developed in a timely and professional manner. The system has been extremely valuable in fine-tuning my interpretive skills and in providing a simple interface to allow the data to be efficiently interrogated. My experiences to date are as follows:

- Client/Student Interaction: This has been very positive and professional throughout the period of this project. The student took on board all comments and suggestions and worked systematically to develop the MCRDR system to meet specific requirements. This was greatly facilitated by the students' willingness to listen and his ability to effectively communicate using non-technical language.
- MCRDR Development: The software was initially difficult to use because of several 'bugs' and the inevitable requirement for precision in setting attribute ranges within rules to establish classifications. There was also an initial learning period to familiarise myself with each screen, particularly, the 'Similarity Screen' which provides the capacity to interrogate the data. However, the system was very quickly debugged and with some practice and assistance I was able to define rules for specific classifications and effectively use these classifications to interrogate the data.
- Analysis Speed: Before the commencement of this project I had spent many (!) hours interrogating lung function data (over one hundred attributes per patient) using a commercial spreadsheet (Excel). Apart from

being extremely tedious, the use of a spreadsheet to interrogate patterns of lung function was very awkward and prone to human errors when sorting and, in particular when selecting and analysing subsets of data. I was therefore in a position to make direct comparison between the utility of the MCRDR system and spreadsheet to discover new knowledge.

The capacity of the MCRDR system to easily select subsets of data, examine individual data within these subsets, and to export data sets into other programs has greatly improved the speed and accuracy with which data can be interrogated for new knowledge. For example, using a spreadsheet to investigate the effect of bronchodilators (these are inhaled asthma drugs used to dilate the airways) on lung function took me at least 3 hours to complete relatively basic analyses and I often had to repeat parts of the analysis due to errors. However, using the MCRDR system I have been able to painlessly complete the same analysis (including exporting and plotting data) in 15 minutes! This is a 12 fold increase in analysis speed. In addition, the analysis using the MCRDRM system was extremely easy to accomplish in a few keystrokes, and the analysis was far less prone to human error.

An important attribute of the MCRDRM system is that it responds almost instantaneously and any error(s) in data selection due to an inappropriate rule or attribute range is readily apparent and easily corrected. I have found this to be an important feature of the system as it also allows unhelpful analyses to be identified quickly, thus facilitating the ongoing search for new knowledge.

- Application of the MCRDR System to Acquire New Knowledge: I have used the system to interrogate lung function data and will continue to use the system into the future. Three examples are as follows:
    - **Can FVC be used to identify patients with small lungs?** The 'gold standard' lung function index of lung size is the total lung capacity, TLC (the volume of air contained in the lungs at full lung inflation). This index is measured using specialised equipment (i.e. whole body plethysmograph) that is only available in large and well equipped lung function laboratories. However, all laboratories, including many GPs and community clinics, have simple and less expensive equipment to measure the forced vital capacity, FVC (i.e. the maximum volume of air that can blown out). TLC is always larger than FVC because TLC includes the volume of any air that cannot be exhaled and this can be affected by lung diseases such as asthma and smoking related chronic obstructive lung disease (COPD). To answer this question I used the MCRDR system to classify patients into two separate groups: 1) those with a low TLC (i.e. definitely have small lungs), and 2) those with a low FVC (i.e. may have small lungs). Comparison of the two groups showed that using FVC alone to define 'small lungs' accounted for 60% of the patients with a low TLC (i.e. FVC misclassified patients as having small lungs in 40% of cases). Subsequent review of published literature agreed with my analysis. I then proceeded to separately inspect the 'FVC' and 'TLC' datasets and compare these with a new dataset which included patients who met the criteria of both

a low TLC and low FVC, to determine whether the misclassification of 'small lungs' based on FVC alone could be improved by including or excluding other lung function indices in the rule. This work is ongoing, but to date it appears that including patients with a normal index, FET (forced expired time – i.e. how long it takes for the patients to expire fully), to the rule improves the specificity.

- **Correction of VA for Airflow obstruction to Estimate TLC**. Alveolar volume (VA) is an estimate of TLC (defined above). It is obtained by inhaling a single breath of air containing a known concentration of an inert gas such as helium and measuring the degree to which it is diluted with residual air already in the lungs. Although VA is often used as a surrogate of TLC it can substantially underestimate TLC in the presence of diseases causing airflow obstruction (i.e. difficulty blowing out quickly due to narrowed airways – indicated by a low FEV1/FVC ratio). VA underestimates TLC in obstruction because the inert gas cannot penetrate to all lung regions. I used the MCRDR system to separately select and export all patients in the database and separately those who had: 1) no obstruction (normal FEV1/FVC ratio), and 2) obstruction (low FEV1/FVC ratio). The analysis revealed that VA provided a good estimated of TLC in people who were able to blow out quickly, but it underestimated TLC in patients with airflow obstruction. The relationship between FEV1/FVC and the ratio VA/TLC was found to be linear: VA/TLC = 0.413 x FEV1/FVC + 0.57 and could be applied to estimate TLC from measurements of VA. This relationship is similar to published data from one unconfirmed study.
- **Is Obesity Associated with Increased Airflow Obstruction?** Obese people often have smaller than normal lungs because the large abdomen can push the diaphragm into the chest and mass of tissue pressing on the chest can prevent full lung inflation. Although the lungs themselves are often normal, the small lung volume results in narrow airways (airway cross-section is related to lung size) and this may result in difficulty blowing out quickly. I used the MCRDR system to identify normal and overweight people (based on their body mass index) and investigated whether they there was evidence that the speed they could blow out was a function of body mass index. This analysis is ongoing but preliminary data suggests that the incidence of airflow obstruction is not higher in people who are overweight, although they do tend to have smaller lungs. This does not exclude an effect when the patients are sleeping when the weight of the abdomen on the diaphragm is accentuated because lung function tests are performed in the sitting position.

- Overall, the MCRDR system has proved valuable in classifying lung physiology and for rapidly investigating patterns of lung function in health and disease. I believe the system will be invaluable in research, education and auditing, particularly if the following additional facilities are implemented (some of these will be available via Afshin).

Suggestions to Improve the Utility of the MCRDR System:

- Include the mean and standard deviation for data sets ('Included') in the 'Similarity' screen. At present only the 'range' for each attribute is given which provides little information about the distribution of the data.
- Include facility to select patients meeting specifically defined ranges in the 'Similarity' screen. This would enable the data from selected patients to be further investigated, or divided into subsets, without leaving the 'Similarity' screen and entering a new rule. For example, when investigating patients who have small lungs, it would be helpful to be able to select those that also meet additional criteria when looking for similarities.
- More descriptive labelling of icons within each screen to avoid confusion for novices would be helpful, together with a 'help' function.
- When entering an inappropriate rule a prompt appears indicating that the case does not meet the rule. It would be helpful to if this prompt were to give some feedback to the operator indicating which part(s) of the rule was not met. It would also be useful to have the facility to enter a rule without having to first identify a specific case that meets it. This would be useful when enquiring about the frequency that a specific rule is met and also to identify a case which meets it.
- The capacity to combine individual rules would be extremely useful. As I understand it, this can be done only by re-entering a new rule that covers several individual ones that have already been entered.
- The MCRDR system should provide the feedback as to whether any of the cases have not received a classification. At present, their may be cases where none of the rules applies to them.
- For future applications it would be critical to develop an interface for new data to be imported into the MCRDR system (I am not sure whether this is already possible). It is also important that the system be able to cope with missing data as future datasets will inevitably have missing data.
- In the future we wish to expand the database to include additional attributes and to identify when a patient is having repeat tests.
- The ability to easily change the title for each attribute would be useful (this may be possible now). Also, a maths function could also be useful for establishing new derived attributes – these may only become apparent after the MCRDR system is established.
- The MCRDR system could be very useful for assessing a students/professional knowledge to accurately interpret lung function data. Perhaps this could be achieved by including an additional 'Teaching' screen whereby a student/professional would be asked to interpret a number of cases randomly selected to cover a range of rules. The system could then provide a summary score.

Well done.

David P. Johns PhD, FANZSRS