



OpenAIR@RGU

The Open Access Institutional Repository at Robert Gordon University

<http://openair.rgu.ac.uk>

Citation Details

Citation for the version of the work held in 'OpenAIR@RGU':

MASSIE, S., 2006. Complexity modelling for case knowledge maintenance in case-based reasoning. Available from *OpenAIR@RGU*. [online]. Available from: <http://openair.rgu.ac.uk>

Copyright

Items in 'OpenAIR@RGU', Robert Gordon University Open Access Institutional Repository, are protected by copyright and intellectual property law. If you believe that any material held in 'OpenAIR@RGU' infringes copyright, please contact openair-help@rgu.ac.uk with details. The item will be removed from the repository while the claim is investigated.



THE
ROBERT GORDON
UNIVERSITY
ABERDEEN

Complexity Modelling for Case Knowledge Maintenance in Case-Based Reasoning

Stewart Massie

A thesis submitted in partial fulfilment
of the requirements of
The Robert Gordon University
for the degree of Doctor of Philosophy

December 2006

Abstract

Case-based reasoning solves new problems by re-using the solutions of previously solved similar problems and is popular because many of the knowledge engineering demands of conventional knowledge-based systems are removed. The content of the case knowledge container is critical to the performance of case-based classification systems. However, the knowledge engineer is given little support in the selection of suitable techniques to maintain and monitor the case base. This research investigates the coverage, competence and problem-solving capacity of case knowledge with the aim of developing techniques to model and maintain the case base.

We present a novel technique that creates a model of the case base by measuring the uncertainty in local areas of the problem space based on the local mix of solutions present. The model provides an insight into the structure of a case base by means of a complexity profile that can assist maintenance decision-making and provide a benchmark to assess future changes to the case base.

The distribution of cases in the case base is critical to the performance of a case-based reasoning system. We argue that classification boundaries represent important regions of the problem space and develop two complexity-guided algorithms which use boundary identification techniques to actively discover cases close to boundaries. We introduce a complexity-guided redundancy reduction algorithm which uses a case complexity threshold to retain cases close to boundaries and delete cases that form single class clusters. The algorithm offers control over the balance between maintaining competence and reducing case base size.

The performance of a case-based reasoning system relies on the integrity of its case base but in real life applications the available data invariably contains erroneous, noisy cases. Automated removal of these noisy cases can improve system accuracy. In addition, error rates can often be reduced by removing cases to give smoother decision boundaries between classes. We show that the *optimal* level of boundary smoothing is domain dependent and, therefore, our approach to error reduction reacts to the characteristics of the domain by setting an appropriate level of smoothing. We introduce a novel algorithm which identifies and removes both noisy and boundary cases with the aid of a local distance ratio.

A prototype interface has been developed that shows how the modelling and maintenance approaches can be used in practice in an interactive manner. The interface allows the knowledge engineer to make informed maintenance choices without the need for extensive evaluation effort while, at the same time, retaining control over the process. One of the strengths of our approach is in applying a consistent, integrated method to case base maintenance to provide a transparent process that gives a degree of explanation.

Contents

1	Introduction	1
1.1	Case-Based Reasoning	2
1.1.1	The CBR Cycle	3
1.1.2	Key Assumptions Underlying CBR	5
1.1.3	CBR Knowledge Containers	6
1.1.4	CBR Performance Considerations	7
1.2	Motivation behind the Research	8
1.3	Research Objectives	9
1.4	Thesis Overview	9
2	Literature Survey	11
2.1	Competence	11
2.1.1	Evaluation Methods	12
2.1.2	Competence Models	13
2.2	Case Base Maintenance	19
2.2.1	Case Reduction	20
2.2.2	Case Discovery	29
2.3	Case Visualisation	32
2.3.1	Scatter Graphs	33
2.3.2	Force-Directed Graphs	34
2.3.3	Parallel Co-ordinate Plots	35
2.3.4	Visualisation Summary	36

3	Problems with Existing Case Base Models	38
3.1	Evaluation of Existing Case base Models	40
3.1.1	Evaluation	41
3.1.2	Discussion	43
3.2	Issues with Classification Problems	48
3.2.1	Inherent Problem Complexity	48
3.2.2	Cases Available	50
3.2.3	Discussion of Classification Issues	52
3.3	Chapter Summary	54
4	Complexity Model for Classification Problems	55
4.1	Boundary Approach	56
4.1.1	Boundary Measures	56
4.1.2	Experimental Evaluation of Boundary Approach	59
4.2	Complexity Approach	63
4.2.1	Case Complexity	63
4.2.2	Complexity Profiling	66
4.3	Chapter Summary	70
5	Complexity-Guided Case base Maintenance	72
5.1	Case Discovery	73
5.1.1	Complexity-Guided Approach	75
5.1.2	Creating a new Case	78
5.1.3	Algorithms	80
5.2	Redundancy Reduction	81
5.2.1	Complexity-Guided Approach	83
5.2.2	Complexity Threshold Editing	86
5.3	Error Reduction	87
5.3.1	Profiling to Identify Harmful Cases	89
5.3.2	Complexity-Guided Error Reduction	93
5.3.3	Threshold Error Reduction Algorithm	94
5.4	Maintenance Algorithms in Practice	95

5.5	Chapter Summary	102
6	Evaluation	104
6.1	Datasets	104
6.2	Complexity Model	105
6.3	Complexity-Guided Case Discovery	107
6.3.1	Experimental Design	107
6.3.2	Results and Discussion	110
6.3.3	Case Discovery with Noise Filtering	112
6.4	Complexity-Guided Editing	115
6.4.1	Experimental Design	116
6.4.2	Results and Discussion	117
6.5	Error Reduction	119
6.5.1	Initial Experiments	120
6.5.2	Experiments on Datasets with Artificial Noise	121
6.6	Chapter Summary	122
7	Conclusions and Future Work	124
7.1	Achievements and Contributions	124
7.2	Going Forward	127
7.2.1	Shortcomings in Approaches	128
7.2.2	Limitations of Scope	129
7.3	Final Comments	131
A	Published Papers	143

List of Figures

1.1	Classic CBR cycle	3
2.1	Graph showing calculation of coverage and reachability sets	17
2.2	Classification case reduction algorithm	21
2.3	Case Coverage and Classification	27
3.1	Typical flat attribute/value representation of a case base	39
3.2	Graph showing accuracy on test set and competence prediction for different case base sizes on four datasets	43
3.3	Relationship between case density and number of cases	45
3.4	Visualisation of two different case bases for the same problem-solving domain	46
3.5	Visualisation of two problems with similar case base composition but different boundary conditions	47
3.6	Effect of boundary overlap on accuracy in classification tasks	49
3.7	Effect of the number and position of cases on classification complexity . . .	51
3.8	Effect of noise on problem complexity	51
3.9	Typical graph of test set accuracy as a case base grows	52
4.1	Identification of boundary cases within a cluster	57
4.2	Identification of boundary cases from opposing group	57
4.3	Identification of individual decision boundaries	58
4.4	Graph of an artificial dataset identifying the decision boundary and separation between classes	59
4.5	Graphs of leave-one-out accuracy and the three boundary metrics on an artificial dataset with varying boundary separation	60

4.6	Class composition of a cases's neighbourhood	64
4.7	Proportion of case c_1 's neighbours belonging to a different class	65
4.8	Typical graph of local complexity profile	67
4.9	Typical complexity profile for an edited case base	68
4.10	Complexity profiles for sample datasets	69
5.1	Typical graph of test set accuracy as a case base grows	74
5.2	A target case's nearest unlike neighbour being identified	76
5.3	Target cases identified without clustering	77
5.4	Target cases identified with clustering	77
5.5	<i>Friend to enemy</i> distance ratio	79
5.6	Illustration of COMPLEXITY	80
5.7	Illustration of COMPLEXITY+	81
5.8	Editing algorithms strike a balance between conservative and aggressive editing	83
5.9	Basic threshold approach to redundancy reduction illustrating the impact of different thresholds on edited case base size	84
5.10	Illustration of Iris dataset highlighting need for refinements in relation to mutual redundancy and noise	85
5.11	Complexity threshold editing algorithm	87
5.12	Calculation of friend:enemy ratio	89
5.13	Typical graph of Friend:Enemy ratio profile	91
5.14	Sample profiles for three classic dataset	92
5.15	Accuracy of edited case bases as cases with ratio above threshold are removed	94
5.16	Threshold error reduction algorithm, TER	96
5.17	Prototype interface for complexity-guided maintenance prior to parameter setting	97
5.18	Prototype interface for complexity-guided redundancy reduction showing the selected threshold and cases proposed for deletion	98
5.19	Prototype interface for complexity-guided case discovery	99
5.20	Prototype interface for complexity-guided error reduction	100
6.1	Error rate correlation	107

6.2	Noise level correlation	107
6.3	Illustration of COMPETENCE	108
6.4	Accuracy of growing case bases as cases are discovered	109
6.5	Accuracy as cases are discovered with COMPLEXITY with differing noise filtering levels	113
6.6	Accuracy as cases are discovered with COMPLEXITY+ for varying noise filtering levels	114

List of Tables

6.1	Comparison of UCI datasets used for evaluation	105
6.2	Results summary of complexity profile indicators compared to alternative measures	106
6.3	Results summary according to significance.	111
6.4	Comparison of average test set accuracy for alternative editing algorithms .	117
6.5	Comparison of edited case base size for alternative editing algorithms . . .	117
6.6	Comparison of average test set accuracy	120
6.7	Comparison of edited case base size	121
6.8	Comparison of average test set accuracy	122
6.9	Comparison of edited case base size	122

Chapter 1

Introduction

Artificial Intelligence (AI) is a branch of computer science concerned with the development of intelligent behaviour and learning in machines. One of the initial goals was to create machines that could match or exceed human problem-solving abilities. Active efforts to achieve this goal began in the 1950's (Turing 1950, McCarthy et al. 1955) and, while computers may exceed human problem-solving capabilities in many specific tasks, this goal is still a long way off in many problem areas today.

In broad terms, case-based reasoning (CBR) is the process of solving new problems based on the solutions of similar past problems (Kolodner 1993, Riesbeck & Schank 1989). CBR is a branch of AI in which reasoning is based on previous experiences. These experiences are stored as problem-solving instances, called cases (Kolodner 1983), and generally include a description of the problem faced and of the solution applied and may also include a measure of the success rate of the solution. This makes CBR different from other AI approaches which rely on generalised knowledge of a problem domain. An advantage of CBR is that knowledge acquisition in the form of experiences is often easier to gather than complex domain specific knowledge possibly in the form of rules. CBR eases the *knowledge elicitation bottleneck* that hinders the development of expert systems (Gonzalez & Dankel 1993, Giarratano & Riley 1994).

A criticism of the CBR approach is that anecdotal experiences are accepted as its main source of knowledge and without validation of their quality there is no guarantee

that generalizations made from them will be correct. While the quality of the store of cases does control the performance of a CBR system, maintenance of the case knowledge is one approach that can alleviate this potential problem.

The focus of this research is on maintenance of the reasoner's experiential knowledge source, called the case base. Maintenance of the case base will be made with some specific performance objective in mind. The full CBR process plays an integral role in the overall system performance, hence, CBR in general requires some consideration at this stage. This section gives background information on CBR, looks briefly at the motivation behind this research and then details the research objectives for this work.

1.1 Case-Based Reasoning

The CBR paradigm is simple to understand largely because it emulates an approach people use when faced with problem-solving situations in their everyday lives. When a person is faced with a new problem their first approach is generally to think of previous similar problems and to try to use the solutions to those problems, perhaps with some changes, to solve the new problem. CBR solves problems in exactly this way by relying on specific knowledge about previous problems and their solutions.

CBR has its roots in cognitive science research. Schank & Abelson (1977) suggested that our knowledge about a situation is stored in the brain as scripts that capture information about recurrent activities. Schank's (1982) dynamic memory proposed a structure called memory organisation packets (MOPs) that are vital in problem-solving and learning. MOPs hold situation patterns as a general model and specific experiences as a specialisation of these with the two being linked through an indexing web. In a similar approach, CBR systems often use indexing to partition the case base and cases to provide the specialised problem-solving knowledge. CBR is both a cognitive model of how people solve problems and a methodology for building AI systems (Kolodner 1993).

This section looks first at the CBR cycle and the processes involved, before identifying the knowledge a CBR system should contain to allow these processes to take place, and finally looking at performance indicators that can be used to measure how well the CBR system achieves its goals.

1.1.1 The CBR Cycle

Different ways have been proposed to describe the CBR process, see for example Kolodner (1993) and Leake (1996b), but the traditional Aamodt & Plaza (1994) R^4 CBR cycle, described in Figure 1.1, is the most commonly used. The case base stores previously solved problems with their solutions. When a new problem arrives the most similar cases are *retrieved* and their solutions *reused* to provide a proposed solution which may be *revised* after testing to create a final solution. As a final stage the new problem and solution can be *retained* as a new case in the case base, allowing the system to learn new knowledge. The implementation of these four stages will be looked at in more detail.

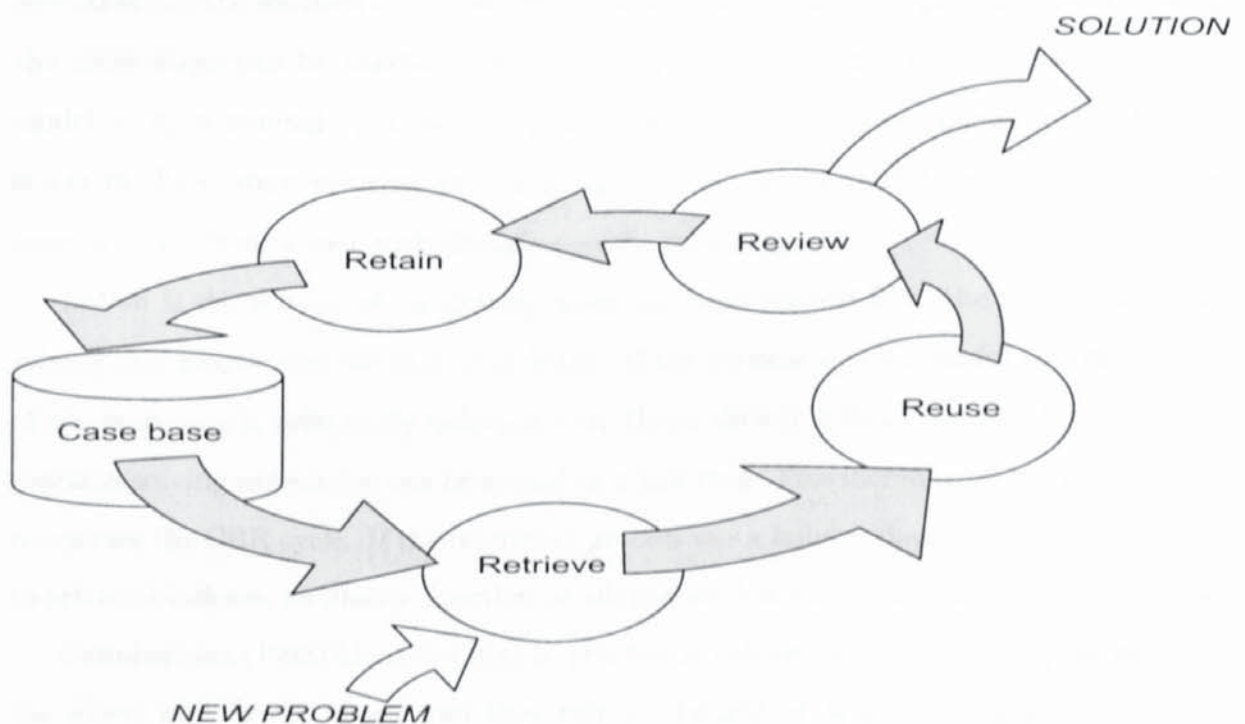


Figure 1.1: Classic CBR cycle

Retrieve is the process of remembering a relevant experience or set of experiences. In order to do this, important information about the current problem must be identified for comparison with the cases in the case base. This comparison or matching is often a two step process. First, a quick search of the cases in the case base is performed, possibly using an index (e.g. C4.5 decision tree index) to filter the cases. A more refined search is then applied, often using k -nearest neighbour (k -NN), to choose either the most similar case or a small set of similar cases. This whole process relies heavily on how cases are

represented and how the similarity measure is defined, but this will be discussed later.

Reuse is applying the solution of the remembered experiences. The retrieved case is checked for consistency with the new problem and the differences identified. In basic classification tasks these differences are often assumed unimportant and the solution class of the retrieved case is re-used as the solution of the new problem. Where a set of cases have been retrieved an average or vote of the retrieved solutions can be used to determine the classification of the new problem. However, systems may have to take these differences into account and modify the retrieved case. This is often referred to as adaptation.

The *revise* stage provides the CBR system with feedback on the quality of the solution by evaluating the solution and if required repairing any fault. The proposed solution from the reuse stage can be validated either by applying it directly, using a domain-specific model or by a manual appraisal. Repair involves identifying, explaining and fixing the errors in the current solution and, although it can also be model-based, often requires manual input from a user with domain-specific knowledge.

Retain is the process of integrating what has been learned from the current problem-solving experience into the case knowledge. If the process was successful and the results of the reasoning is sufficiently different from those already held in the case base, the new problem-solving experience can be stored as a new case. This incremental learning process completes the CBR cycle. If the reasoning process was a failure, changes can be considered to retrieval indexes, similarity function or adaptation knowledge to learn from this failure.

Cunningham (1998) identifies that in practice it can be difficult to distinguish between the *Reuse* and *Revise* stages and they can be thought of as a single *Adaptation* stage. Adaptation is one of the more difficult problems being addressed by CBR research and it has been argued by Barletta (1994) that CBR loses its knowledge engineering advantages if considerable adaptation is required. In fact, most of the successful CBR applications involve no adaptation. An alternative view, is that adaptation should play a greater role in the CBR process. Smyth & Keane (1996), for example, propose that adaptation should be used to guide the retrieval phase.

1.1.2 Key Assumptions Underlying CBR

CBR systems can be built without passing through the knowledge elicitation bottleneck since elicitation becomes a simpler task of acquiring past cases. Hence, CBR is often applied to solve problems where no explicit domain model exists. However, in adopting CBR certain implicit assumptions are made about the domain.

- **Regularity:** this requires that similar problems have similar solutions. CBR systems solve new problems by retrieving similar, solved problems from the case base and re-applying their solutions to the new problem. If, on a general level, the solutions of similar problems do not apply to new target problems then CBR is not a suitable problem-solving approach for the domain. Fortunately, the world is generally a regular place and the fact that *similar problems have similar solutions* tends to apply in many domains. It is our contention that in local areas of the problem space this assumption may not always hold true and that these areas require special consideration. We revisit this assumption in later chapters.
- **Repetition:** Using previous experiences as the basis for problem-solving is only a reasonable approach where similar situations repeat themselves. If they do not, there is no point remembering the experience from a problem-solving perspective. This assumption is fulfilled in many domains where similar problems do tend to re-occur over time e.g. medical or fault diagnosis.
- **Representativeness:** this requires the contents of the case base to be a good approximation of the problems the system will encounter (Smyth & McKenna 1999a). If a case base is not representative of problems to be faced then the CBR system cannot be expected to provide solutions to similar problems.
- **Experiential:** CBR is essentially a memory-based problem-solving approach and, as such, the case base should be the main source of knowledge. If a system relies primarily on domain knowledge for problem-solving and uses experiential knowledge only as a secondary knowledge source (e.g. speed up learners) then it is not a true CBR system.

The extent to which these four assumptions hold true are central to the suitability of CBR as a problem-solver for the domain. In addition, the range of target problems that a CBR system can successfully solve, called competence, is to a large extent determined by the regularity, repetition and representativeness assumptions. Our approach to modelling and maintaining case knowledge focuses on measuring, maintaining or improving competence and, as such, these assumptions also underpin our work.

1.1.3 CBR Knowledge Containers

In CBR, knowledge is information that can be used to help solve a problem. Richter (1998) identifies *vocabulary*, *case base*, *retrieval* knowledge and *adaptation* knowledge as the four main CBR knowledge containers that participate to solve problems in CBR.

- *Vocabulary* refers to the way in which a case is represented. A case, in its simplest form, consists of two parts: the problem and solution. The problem normally consists of a set of attributes and values called an attribute-value vector. The solution can be a single attribute (e.g. a classification) or a more complex structure (e.g. a route plan).
- The *case base* contains the set of domain experiences (cases) used to solve new problems and is usually the key knowledge source within a CBR system. A case captures the problem and its solution.
- *Retrieval* knowledge finds relevant, similar cases by identifying which attributes should be considered when determining relevance. Typically, the attribute value differences between cases are compared and the relative importance of each attribute considered when determining similarity.
- *Adaptation* refers to the knowledge required to transform the solution of the retrieved cases into a solution to the current problem. This is achieved by considering the differences between the retrieved cases and the new problem to tailor the solution to the current problem. This knowledge may take many forms, but often is in the form of rules.

CBR systems provide at least a partial solution to the knowledge acquisition problems associated with rule based systems, by reducing the need for problem analysis (Cunningham 1998). While this may be true for case knowledge, in CBR there is also a need to acquire vocabulary, retrieval and adaptation knowledge before a system is operational.

The knowledge held in a CBR system's knowledge containers should not remain fixed. Numerous internal or external changes can occur that affect the relationship between the system's knowledge and the knowledge required to solve the problems being faced. In order to react to these changes the contents of the knowledge containers must be adjustable. Changing the contents of the knowledge containers is referred to as maintenance and can be controlled by domain experts, however, researchers are looking at automated or semi-automated approaches (Craw, Jarmulak & Rowe 2001, Leake & Wilson 2000, Patterson, Rooney & Galushka 2002, McKenna & Smyth 2001a).

Maintenance knowledge represents a fifth knowledge container required to support the others over the life cycle of a CBR system (Patterson, Anand, Dubitzky & Hughes 2000, Iglezakis, Reinartz & Roth-Berghofer 2004). This research will provide techniques to aid the maintenance of the key knowledge source: the case base.

1.1.4 CBR Performance Considerations

It is important to be able to measure how well a CBR system is performing for a given case base and sequence of problems. Smyth & McKenna (1999b) identify three types of top level goals that can be used to measure a CBR system's performance. These are:-

1. Problem-solving efficiency goals (e.g. average problem-solving time).
2. Competence goals (e.g. the range of target problems successfully solved).
3. Solution quality goals (e.g. the average quality of a proposed solution).

There is often a trade off between these performance goals, e.g. a small case base may achieve efficient problem-solving at the expense of competence and quality, and a balance has to be found. Performance considerations are particularly important where maintenance of the knowledge used by the system is taking place because system performance provides a measure of the effectiveness of these maintenance policies. Case base mainte-

nance strategies should be driven by specific performance goals as well as by constraints on the system design e.g. limits on case base size.

1.2 Motivation behind the Research

CBR systems rely on the contents of the various knowledge containers, as these affect how well a system performs. Explicit or implicit changes in the task, the reasoning environment or the user base may all affect the fit between the state of current system knowledge and the task being undertaken. This may affect the performance of the system in terms of its efficiency, competence and solution quality. Over time, the system's knowledge must be updated in order to maintain or improve performance as changes take place. This maintenance should take the form of support tools that monitor a system's state to determine when and how knowledge should be updated in response to specific performance criteria.

An increase in experience with the deployment of long term CBR systems has led to recognition within the field that maintenance of existing systems is an essential element of an operational CBR system (Leake, Smyth, Wilson & Yang 2001). Maintenance issues affect all stages of a system's life cycle and, as such, maintenance is increasingly becoming a focus of interest within the CBR research community. This is evident by an increasing number of conference papers (Gomes, Pereira, Carreiro, Paiva, Seco, Ferreira & Bento 2003, Patterson, Rooney & Galushka 2003, Woon, Knight & Pedridis 2003, Zehraoui, Kanawati & Salotti 2003) and specialised maintenance workshops (e.g. Workshop on Flexible Strategies for Maintaining Knowledge Containers 14th European Conference on Artificial Intelligence 2000).

In theory, knowledge can be held in any of the knowledge containers and a lack of knowledge in one container can be offset by increasing the knowledge in another. However, as the underlying philosophy of CBR is to reuse previous experiences the case base should provide the main knowledge source. This is essential if the knowledge acquisition problems discussed previously are to be avoided.

This research expands on previous work of Robert Gordon University's CBR group that has looked at learning both retrieval knowledge (Jarmulak, Craw & Rowe 2000) and

adaptation knowledge (Wiratunga, Craw & Rowe 2002).

1.3 Research Objectives

This research set out to develop techniques to maintain the case base of CBR systems. A model of the case base was to be developed and used, in conjunction with techniques that identify gaps in the case base, to provide active learning methods that identify new cases and problems to enhance the problem-solving capability of the CBR system. In addition, the model was to be used to develop case editing algorithms that identify redundant or harmful cases that no longer make a problem-solving contribution and produce an edited set of cases that balance problem-solving cost and loss of accuracy. Incorporating these maintenance techniques into a CBR system was to include a new case base visualization tool, based on the relationship between cases, that aids the case authoring and maintenance processes.

Specifically, five research objectives were addressed:-

1. *Develop a technique to model the problem-solving capabilities of a case base.*
2. *Develop a technique to identify gaps and create new cases to fill them.*
3. *Develop a case base maintenance algorithm that identifies redundant cases.*
4. *Develop a case base maintenance algorithm that identifies harmful cases.*
5. *Create a visualisation tool that demonstrates case coverage and allows the user to view redundancy and gaps.*

1.4 Thesis Overview

This thesis develops an integrated suite of case base maintenance tools that initially provide the knowledge engineer with an insight into the structure and composition of the case base by way of a local complexity model and case profiling. The model is then used to develop case discovery and editing algorithms. An interactive prototype interface is developed to aid the knowledge engineer make informed decisions.

In this Chapter we have given some context to the work that follows with a brief overview of CBR, looked at the motivation behind the research and identified the initial research objectives that have directed the research. Chapter 2 reviews recent work in CBR in relation to modelling of the case base, case discovery and case base editing. Alternative multi-dimensional visualisation techniques are also considered.

In Chapter 3 we evaluate an existing case base competence model on classification problems and consider some of the key issues for creating a model for classification problems. Chapter 4 investigates alternative approaches to case base modelling before developing a new complexity-guided case base model for classification problems. We show how the model can aid the knowledge engineer make informed maintenance decisions.

Chapter 5 introduces three new case base maintenance techniques informed by local information supplied from the complexity model. Our case discovery algorithm uses a complexity metric, boundary detection and clustering to detect areas of the problem space that need the support of new cases and proposes new cases to occupy the space. A case base editing algorithm is developed that removes redundant cases by setting a threshold for case complexities but leaves control of the balance between case base size and competence with the knowledge engineer. A novel error reduction algorithm is introduced in which the extent to which noisy and potentially harmful cases are removed is controlled by a stopping criteria that is adjusted to suit the domain characteristics. Finally, an interactive interface is described that allows the knowledge engineer to make informed maintenance decisions about the case base by use of visualisations to explain the maintenance process and expected results.

In Chapter 6 we evaluate the ability of our complexity model to predict real dataset values for accuracy and noise. The new maintenance algorithms are evaluated experimentally by comparing their performance with relevant benchmark algorithms on UCI datasets.

The conclusions in Chapter 7 summarise the achievements and contributions of the research, discuss some of its limitations, and identify possible extensions and directions of future research.

Chapter 2

Literature Survey

This chapter is a literature review that provides an overview and critical evaluation of recent work conducted in maintenance of CBR systems with an emphasis on competence modelling, case editing, case discovery and visualisation of the case base. Relevant research, in relation to the research objectives, is discussed and failings or gaps are identified that justify the need for this research.

2.1 Competence

Competence is a measure of how well a CBR system fulfils its problem-solving goals. It is a fundamental evaluation criterion of a CBR system or, in fact, of any problem-solving system. Competence is difficult to measure because, while there may be some general information about the purpose of the system and the type of problems it will face, the exact problems are not known in advance. CBR is primarily a problem-solving methodology and, as a consequence, competence is usually taken to be a system's ability to solve unseen problems correctly, although other measures are also possible. Indeed much recent research has looked at diversity as one of the possible goals of CBR (McCarthy, Reilly, Smyth & McGinty 2005, McSherry 2002). In this project competence will be taken to mean a system's ability to solve problems and will be measured as the proportion of problems faced that it can solve successfully regardless of how good the solutions are or how fast they are produced.

There has been limited research aimed at modelling competence to predict a system's

performance and to provide more informed knowledge on the value of each case. In this section, research on modelling will be discussed but first the methods typically used to evaluate a case base empirically will be reviewed.

2.1.1 Evaluation Methods

The problems that a CBR system will be asked to solve are unknown in advance. Evaluation methods overcome this by making the assumption that the case base is a representative sample of problems that will be faced. This representativeness assumption is reasonable because a CBR system could not possibly be a good problem solver if the case base were not representative. However, the results will only provide a good estimate of competence if this assumption holds. All the methods discussed here split the case base into a training set and test set and calculate a success rate (proportion correctly solved) for the test set. The existing data is being used to give an estimate of future performance and will only give a reasonable estimate if the test data is not used to train the CBR system in any way.

Three evaluation methods are discussed briefly here but are covered in more detail in (King, Feng & Sutherland 1995, Mitchell 1997, Witten & Frank 2000) including advice on deciding which method is most appropriate for different situations.

The Training and Test Set evaluation method splits the case data into a case base (typically two-thirds of the data) and a test set (the remainder of the data). The test set, consisting of cases that were not used to train the system, is used to evaluate the performance of the CBR system in this approach. It tends to be used where case data is plentiful. One problem with this approach is that the sample used for testing may not be representative. The chance of this happening can be reduced in classification problems by using a procedure called stratification in which we ensure that each class is proportionally represented in both the case base and the test set.

In cross-validation, the case data is split into a fixed number (n) of equal sized partitions or folds, giving n -fold cross-validation. One partition is used for testing, with the remaining $n-1$ partitions being used as the case base. The procedure is repeated for each partition so that each case has been used exactly once for testing. The success rate across all the partitions is averaged to give an overall success rate or competence. Stratified n -fold cross-validation is the standard evaluation technique in situations where limited case data

is available. The process can be repeated to give more reliable results.

Leave-One-Out evaluation is a special case of n-fold cross-validation where n is the total number of cases available. Each case is used in turn singly as the test set, with the remainder of the cases being the case base. The proportion of cases correctly solved gives the success rate. This method allows the greatest amount of training data to be used for evaluation and will give the same result every time as no random sampling is involved. It has the disadvantage of high computational cost.

In CBR research, case base evaluation is usually performed using the Training and Test set method. Cross-Validation techniques are more computationally expensive but they are more accurate and should probably be the preferred choice where the evaluation cost can be justified and data is not plentiful. If data is limited Leave-One-Out testing could be used.

The evaluation methods described here are well established methods used across a range of research disciplines. They are important to this research for two reasons. Firstly, to provide a means to validate the output of the models being developed, but also because some of these techniques, in particular leave-one-out testing, are used as part of the modelling process itself.

2.1.2 Competence Models

A model is a representation of reality that helps someone understand, manage or control that reality (Ackoff & Sasieni 1968). Models are always a simplification and it is this simplification that makes them useful. Therefore, an important question is "what degree of simplification is sensible?".

Modelling the competence of a CBR case base is the process of representing the case base in such a way that it gives an estimate of competence, while at the same time providing information to allow the knowledge engineer to manage the case base. Evaluation methods give a good estimate of competence, however, in addition to being time consuming, they give little information on the structure of the case base or the competence in local areas of the case base. Modelling aims to provide this additional information, in order to develop informed case base maintenance policies.

Two related concepts are used to model problem-solving ability: *coverage* and *compe-*

tence. The coverage of a system is the ratio of possible problems a system can face that it can solve successfully. In contrast, competence is the proportion of problems that it will actually face that it can solve successfully (Smyth & McKenna 1999a). Competence gives the better measure of how a CBR system will perform in real situations but the problems a system will face are unknown in advance.

Coverage Models

The coverage approach is based on the assumption that the problem space is finite and attempts to measure the number of points within this problem space that are covered by the case base. The number of cases in the case base does not give a realistic measure of coverage as there may be redundant cases occupying the same point in the finite problem space. A lower bound on the coverage can be found by finding the number of different points covered by the case base. McSherry (1999) adopts this empirical approach to coverage. A case is considered to be a vector of nominal attribute values and the total size of the problem space is the Cartesian product of the domains of these attributes. An algorithm called *disCover* is used to identify all the points within the problem space that a given case base can solve. This algorithm looks at a point in the problem space and if an exact match does not exist attempts to find other cases that provide a solution for this location using a linear adaptation function.

This empirical coverage approach only applies where the cases are represented in a finite problem space. Many case representations use numeric attributes or more complex representations. Even within a finite problem space, as the number of attributes increases, the size of the problem space can become very large, making this a computationally expensive algorithm. The use of a linear adaptation function is also rather limiting and could not be applied to most CBR systems to give a reasonable measure of which target cases can be solved. Another problem of this approach is that cases which can exist in theory as a given combination of attribute values may not be possible in reality, as the combination of values is impossible. This problem is addressed by a rule-based approach (McSherry 2000) that identifies points in the problem space that represent invalid attribute value combinations and excluding these from the coverage calculation. However elicitation of this rule-based domain knowledge adds extra expense to the models development.

Competence Models

The competence model approaches are based on the same assumption as the empirical evaluation methods: namely that the case base is a representative sample of the problems to be faced. This representativeness assumption removes the problem of a very large or infinite target problem space encountered in the coverage approach. Competence models aim to analyse the case base itself rather than test it against existing data to answer the question: "How good is the case base?" Competence depends loosely on properties like the number and density of cases. However, since competence is concerned with the range of target problems that a given system can solve, it also depends on the problem-solving ability of the system and must involve the retrieval process and adaptation knowledge of a system. The number and density of cases can be measured but calculating the problem-solving ability of a case, in terms of its retrieval and adaptation characteristics is not so simple.

Smyth & Keane (1995) create a competence model that uses the case base to simulate the full domain and leave-one-out testing to measure the problem-solving ability of each case by assembling two important performance indicators: *coverage* and *reachability*. The coverage of a case is the set of problems it can solve, conversely, reachability is the set of all cases that can solve it. In this model *can solve* identifies whether a case can be adapted to solve another and is not based on the similarity between cases. In broad terms coverage and reachability gives a measure of a case's importance and uniqueness respectively.

One problem with the Smyth & Keane (1995) model is that it does not fully consider the interaction between cases, for example, where many cases can solve the same target problems. Smyth & McKenna (1998) extend the model in a finer grained approach that considers the interaction between cases by forming independent clusters of cases. The model predicts competence using the following four stages.

Stage 1 - Measure the coverage and reachability of each case.

Stage 2 - Form clusters of cases, called competence groups, using their reachability and coverage sets to group cases that have overlapping sets. Overall system competence is not simply the sum of the individual case coverage sets because there is interaction between cases. Competence groups cluster these interacting cases together to allow

the effect of the group on overall competence to be evaluated. The number of competence groups formed and the number of cases in each depends on four factors: the number of cases in the case base, the density of cases, the retrieval mechanism and the adaptation mechanism.

Stage 3 - The coverage of each competence group is then measured by the group size and density of its cases. It is directly proportional to the group's size and inversely proportional to the group's density and is defined as:

$$\text{GroupCoverage}(G) = 1 + [|G| \cdot (1 - \text{GroupDensity}(G))]$$

The $\text{GroupDensity}(G)$ of a group of cases G is the average case density of the individual cases within the group; where the case density of a case is the average similarity distance to the other cases in the group. Similarity between cases is taken to be a value between 0 and 1 inclusive. The effect of this calculation is a group coverage value highly dependent on the number of cases in the group - but this is not necessarily a realistic measure of coverage. The problem-solving ability of individual cases needs to be given more consideration.

Stage 4 - Calculate the overall competence of the case base. As the competence groups are independent the overall system competence can be calculated directly from the contribution made by each group, and is simply the sum of the coverage of each group. The resultant value has little meaning in its own right, being a value between 0 and the number of cases in the case base plus the number of competence groups.

McKenna & Smyth (2001b) present a modification to the above model that claims to increase its effectiveness. Stage 3 of the original model, measuring the competence of each group, is replaced by a new calculation. First a competence group footprint, a subset of the group's cases whose coverage sets cover all the group's cases, is formed. The coverage of a group is then measured by summing the relative coverage values of its footprint cases. Where relative coverage of a case is defined as the sum of the inverse of the size of the reachability sets of the cases in its coverage set. This model removes the

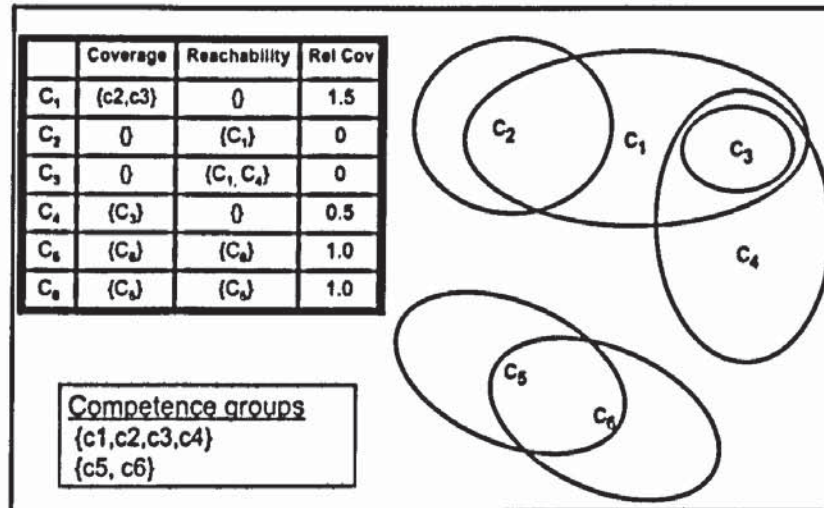


Figure 2.1: Graph showing calculation of coverage and reachability sets

use of a domain specific similarity metric. It is an improvement over the original model because it is less reliant on the number and density of cases and gives more consideration to the problem-solving ability of individual cases.

A simple example showing five cases forming a simple case base is illustrated in Figure 2.1. A case's coverage is shown by an ellipse, for example, case c_1 covers (can solve) both cases c_2 and c_3 . It can be seen that two independent competence groups are formed. The table shows the coverage and reachability set together with the relative cover for each case.

It is claimed that while the modelling of the case base is expensive it is a one time set up cost that can be absorbed by the system. This may be true for relatively stable case bases but many systems use large dynamic case bases with a constant stream of new case additions. While a heuristic approach is proposed to update case classifications for case additions and deletions (Smyth & Keane 1995), it would be difficult to maintain an accurate model in a dynamic environment.

This model relies heavily on adaptation to reflect an idealised competence based on "problems that a system *can* solve". In effect, the cases are partitioned into adaptable and non-adaptable cases with all adaptable cases being treated as equivalent. Adapting cases requires extensive knowledge engineering and can be an expensive process. In practice, processing constraints during the retrieval and adaptation stages of a CBR cycle limits the case base competence. More recent research on case base maintenance algorithms (Zhu

& Yang 1999, Portinale, Torasso & Tavano 1999) takes account of these limitations by setting an adaptation effort threshold which results in a case being considered able to solve a problem only if it can be adapted within a set number of adaptation steps. Defining case coverage in terms of cases that can be solved within the adaptation threshold, limits the idealised competence by considering processing constraints and placing an upper bound on the required adaptation effort.

Leake & Wilson (2000) identify competence as only one aspect of the performance of a case base. A small competent case base may have quick retrieval times but this may be counterbalanced by increased adaptation costs or decreased quality. Smyth & Cunningham (1996) agree that to meet the overall performance goals a CBR system may require balancing trade-offs between competence, quality and efficiency. This requires consideration of processing constraints, retrieval and adaptation processes as well the knowledge containers described earlier. An overall performance criterion should play a direct role in decisions about the CBR system and the case base.

The models discussed so far have largely been applied to recommendation, estimating, design and planning tasks in which adaptation knowledge is used and often the similarity function is related to the number of adaptation steps required. Many CBR problems are classification problems involving no adaptation. Wilson & Martinez (1997) have applied aspects of Smyth and Keane's model to case base editing algorithms for classification tasks. They use the *k*-nearest neighbours and a set of associates to a case which are analogous to Smyth and Keane's reachability and coverage sets. These sets are used to value local case competencies and identify a case's importance in their editing rules. Brighton & Mellish (2001) apply a similar approach to classification problems. The difference being that a case's reachability set is not fixed in size but is bound to include only the nearest neighbours upto the first case belonging to a different class.

All the competence models discussed in this section have been successful in providing local problem-solving information to help inform a range of case base maintenance policies discussed in the next section. These models provide an informed clustering of cases using the problem-solving ability of each case. However, the assumption that these competence groups or clusters are independent is an oversimplification. Failures in a CBR system's ability to solve problems are most likely to occur where different clusters are competing

to solve the new problem i.e. where the new problem lies on a boundary or between competence groups. How the groups interact, when solving new problems, will have a great effect on the competence of a system. These important interactions are ignored by the models.

2.2 Case Base Maintenance

A CBR system relies on the contents of the knowledge containers to achieve its problem-solving ability. These include the case base itself, the similarity measures used to retrieve cases and the adaptation knowledge used to transform cases into a solution. Over time a system's knowledge, task, environment or user-base can change (Leake et al. 2001) and, as a result, knowledge containers may need to be updated to help maintain or improve performance. CBR maintenance refers to the strategies used, and the process of maintaining these knowledge containers, and is essential in order to sustain and improve a CBR system's performance. Although maintenance can address many possible performance goals, it should be aimed at a specific set of performance objectives.

This section looks, in particular, at strategies for maintaining the case base, which is the key knowledge container underpinning a system's performance. Case base maintenance (CBM) is the process of changing or refining the case base and has been defined as follows:

“case base maintenance implements policies for revising the organisation or contents (representation, domain content, accounting information, or implementation) of the case base in order to facilitate future reasoning for a particular set of performance objectives.” (Leake & Wilson 1998)

Hence, if a vast amount of case data is available either initially or as extra cases are added during problem-solving, it is helpful to select only useful examples and ignore redundant cases. If cases are scarce, all available examples are used as case knowledge, but it is also useful to identify gaps in the case knowledge and generate new cases to fill them. CBM involves developing algorithms to accomplish or assist in these case reduction and case creation tasks. The following sections look at these areas in more detail.

2.2.1 Case Reduction

There is an increasing use of large case bases in CBR systems. Over time the case base gets larger, often as a result of indiscriminate storage of cases during the retain stage of the CBR cycle. The cases may be redundant and provide no improvement in competence or may even be harmful, noisy cases that result in a reduction in competence. In either case the inclusion of additional cases will increase storage requirements and retrieval times. The cost of retrieval can grow to the extent that it outweighs the efficiency benefit of additional cases. This is called the *utility problem* (Francis & Ram 1993, Smyth & Cunningham 1996). Research to control case base growth has focused on case base editing (case deletion or selection policies) although some creative, generalisation approaches have also been considered. Case reduction is used in this report to describe both the editing and generalisation approach to the reduction in the case base size.

There are a number of features that distinguish one case reduction algorithm from another and these will provide a framework for the classification and discussion of specific algorithms.

- **Noise Reduction or Compaction.** Noise reduction algorithms aim to improve competence by removing noisy cases. In contrast compaction algorithms, in general terms, aim to achieve the smallest case base size that retains the original competence by removing redundant cases.
- **Selection or Generalisation.** Selection approaches retain existing cases in an edited case base. Whereas, generalisation approaches either create new prototype cases by merging existing cases or partitioning the problem space based on the position of the existing cases. This report concentrates on reduction approaches that retain existing cases as this is more typical of the CBR approach.
- **Search Direction.** Forward selection or incremental approaches start with an initially empty set and add cases to it, whereas backward elimination or decremental approaches delete cases from the initially complete case base.
- **Border or Central Points.** Algorithms can be distinguished on whether they retain central cases or border cases. The reasoning behind keeping border cases is

that central cases do not affect the decision boundaries and can be removed with little effect on classification. However, it can take a large number of border cases to define a decision boundary so some algorithms aim to retain central cases that are more typical of a particular class.

- **Stopping Criteria.** Most algorithms set their own informed stopping criteria but in some the number of cases in the reduced case base can be controlled.

There are a large number of case reduction algorithms and trying to resolve which one to use for any particular set of data is a difficult task. This report simplifies this task by categorising them, see Figure 2.2, using a simple taxonomy based on the features discussed above. The algorithms are first split by whether their main objective is noise reduction or compaction and then secondly on whether an editing or generalisation approach is being used. A final categorisation is made on search direction. The other categories, not used in this taxonomy, provide a framework for discussion of specific algorithms.

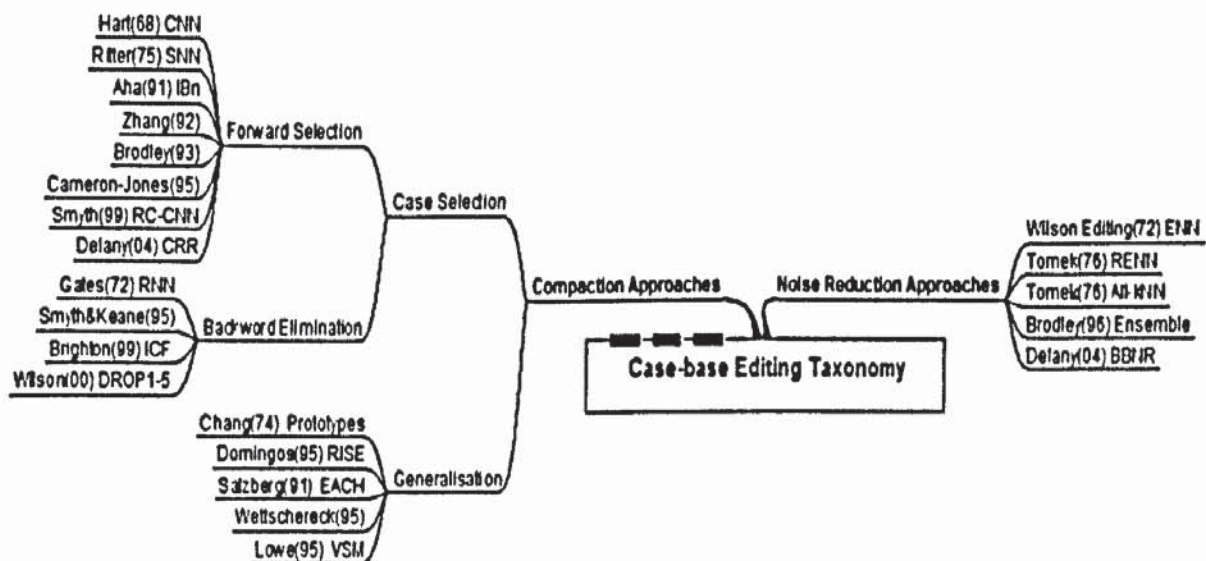


Figure 2.2: Classification case reduction algorithm

Noise Reduction Algorithms

Noise reduction algorithms aim to improve competence by removing cases that are thought to have a detrimental effect on the accuracy of the CBR system. These may be corrupt

cases whose solution is incorrect or, alternately, they may simply be cases whose inclusion in the case base results in other cases being incorrectly solved. These algorithms result in only a few cases being removed and all those discussed here use backward elimination approaches. The Wilson Editing algorithm (Wilson 1972) and its extensions described below are the most notable aimed primarily at noise reduction.

Wilson Editing, also called ENN, attempts to remove noise by considering each case in the case base and removing it if it is incorrectly classified by its k nearest neighbours (with k typically 3). This removes noisy cases but also deletes some cases lying on a boundary between two different classes leaving smoother decision boundaries. This can be thought of as a smoothing of the problem space. Tomek (1976a) extends ENN with the Repeated Wilson Editing method (RENN) and the All k -NN method. RENN extends ENN by repeating the deletion cycle until no more cases are being removed. The All k -NN is similar, only after each iteration the value of k is increased. In some case bases competence is improved by applying these algorithms but there is not a large reduction in the case base size as only noisy and some boundary cases are removed.

Brodley & Friedl (1996) use an ensemble of different type classifiers and use the uncertainty within the results to inform a noise reduction filtering algorithm. Each case is classified using a cross-validation technique and, where a case is misclassified and there is a consensus among the ensemble a misclassified case is removed. If a case is correctly classified or if there is uncertainty in the classification, i.e. no consensus, the case is retained.

Delany & Cunningham (2004) takes a different approach to noise reduction with their blame-based noise reduction algorithm (BBNR) by identifying cases that cause other cases to be misclassified rather than removing cases that are themselves misclassified. The approach extends Smyth & Keane's (1995) model with the introduction of a *liability set*. Leave-one-out testing is used to identify cases that cause other cases to be misclassified and build a case's liability set (cases where this case contributes to a misclassification). Where a case causes more cases to be misclassified than correctly classified the case is removed. This is a conservative approach resulting in the removal of fewer cases.

It is important to remember that these algorithms cannot differentiate between noise and genuine class exceptions. The approaches will also only work where there is a small amount of noise, as a large proportion of noise will no longer appear as exceptions. Hence

careful consideration should be given to the domain and structure of the case base before applying these algorithms to ensure there is a need for noise removal.

Selection Compaction Algorithms

Some early research (Markovich & Scott 1988) advocated a random deletion policy which, although simple, is domain independent and takes no account of the importance of a case. A slightly more reasoned approach (Minton 1990) looked at how often a case was used and deleted those cases that were not accessed frequently. The problem with both approaches is that important cases can be deleted. This has led to numerous compaction strategies that aim to retain the important cases, the most significant are discussed below. These approaches are classified based on their search direction as either forward selection or backward elimination.

- **Forward Selection.** These algorithms start with an empty edited case base and add cases to it by selecting cases from the original case base. Hart's (1968) Condensed Nearest Neighbour rule (CNN) was an early attempt at finding an edited case base that retained important cases and all the algorithms in this section are extensions of it to some extent.

CNN starts with an empty edited case base and randomly selects one case belonging to each output class. Cases are tested in a random order and cases not solved by the edited set are added to it. It proceeds in an iterative fashion until all cases remaining in the original case base are correctly solved by the edited case base. This algorithm is incremental giving it the advantage that additional cases can be added over time as they become available in the retain stage. The problem is that the resulting edited case base is highly dependent upon the order in which the cases are considered, with early cases having a high probability of being included. These early decisions are made on little information and results in an edited case base that is unlikely to be of minimal size. The algorithm also retains noisy cases, as these are cases that are unlikely to be solved by the edited case base.

Ritter, Woodruff, Lowry & Isenhour's (1975) Selective Nearest Neighbour Rule (SNN) extends CNN by including a case in the edited set where it is closer to a

case in the original case base of the same class than to any case in the edited case base of a different class. This improves on CNN by ensuring a minimal consistent subset is found and ensures smoother boundaries by including fewer boundary cases. However, it is computationally more complex than most other algorithms to implement.

The CNN derived algorithms discussed so far are all very sensitive to noise. Aha, Kibler & Albert (1991) developed the IB_n range of algorithms of which the case editing algorithm IB_3 addresses the problem of noise by augmenting CNN with a post processing “wait and see” policy for deleting noisy cases. This is done by keeping a statistical record of how competent the stored cases are at classifying. Noisy cases are likely to be poor classifiers, so stored cases that misclassify on a statistically significant level are removed. In experimentation IB_3 achieves a greater reduction in the number of cases stored and also achieves a better classification accuracy than the other forward selection algorithms discussed so far. A number of researchers have augmented the IB_n algorithms (Cameron-Jones 1992, Zhang 1992, Brodley 1993). IB_3 and these extensions offer an incremental approach that is not too sensitive to noise but, like the other forward selection algorithms, is still sensitive to the order of case presentation.

Several algorithms tackle this presentation order problem by examining the whole case base, in a preprocessing step, to rank the cases prior to applying a forward selection approach, e.g. (Tomek 1976b). This approach can greatly improve performance but results in an algorithm that is not truly incremental and new cases can no longer be added over time. Smyth & McKenna (1999a) present their RC-CNN algorithm extending CNN based on their competence model, discussed in Section 2.1.2. The cases in the original case base are ranked according to the relative coverage metric before applying the CNN algorithm. This ranks cases, for presenting to the CNN algorithm, based on the number of other cases they can solve. However, the algorithm is still sensitive to noise and will tend to select central points first before considering border cases.

Delany & Cunningham (2004) extend the Smyth & McKenna (1998) model to de-

velop a forward selection compaction algorithm. They identify that excessive compaction of the case base can harm the competence of a system and introduce Conservative Redundancy Reduction (CRR), an algorithm that is less aggressive than other state-of-the-art approaches. Starting with an empty edited set, cases with small coverage sets are presented first to be added to the edited set. For each case added to the edited set, the cases in its coverage set are removed from the training set. By selecting the case with small coverage first this algorithm retains boundary cases but gives a more conservative reduction in the case base size.

In general, deleting cases in a CBR system gives some concerns as the cases represent the key knowledge source, and removing cases results in a loss of knowledge to the system. Indexing can provide an alternative approach that, while it does not reduce storage requirements, can improve efficiency while maintaining competence. Smyth & McKenna (1999b) present an interesting combined approach by using the cases retained in their RC-CNN algorithm as an index for the original case base.

The forward selection algorithms have the advantage that they are incremental and can be applied to new cases as they arrive. They are simpler, computationally, as comparisons are made with the smaller edited set rather than with the whole case base. However, they have the problem that they are very sensitive to the order cases are presented and tend to retain noise. Extensions to the basic CNN algorithm address the ordering problem, by ranking the cases, and the noise retention problem, by applying a post-processing analysis. These extension provide very effective case base editing algorithms, however, they also remove some of the advantages of forward selection algorithms.

- **Backward Elimination**

These algorithms start with the complete case base and delete cases using some informed criteria. Gates's (1972) Reduced Nearest Neighbor Rule (RNN) was an early backward elimination approach that starts with an edited case base identical to the original. Cases are deleted from the edited case base if their removal does not cause any other case in the original case base to be misclassified by the remaining cases in the edited case base. It is computationally more expensive than CNN

but experimentation shows it produces smaller edited case bases with improved classification accuracy. Since the instance being removed is not guaranteed to be classified correctly it is less likely to retain noisy cases.

Wilson & Martinez (1997) introduce their Reduction Technique range of case reduction algorithms called RT1-3. These are hybrid algorithms that aim to achieve both noise reduction and compaction. However they are classified here, by their main objective, as compaction algorithms. RT1 is the basic removal scheme. The set of k nearest neighbours and the set of *associates* are determined for each case. The *associates* of a case are the set of cases which have that case as one of their nearest neighbours and is analogous to Smyth & Keane's coverage set introduced in their competence model. RT1 removes a case if at least as many of its associates would be classified correctly without it. This removes noisy cases as their associates are less likely to be misclassified without them. It also removes cases from the centre of clusters. RT2 includes two extensions. First the deletion decision depends on a case's original set of associates in order to improve noise removal. Secondly the cases are ordered by their distance from a case of another class. Those cases farthest from an *enemy* are considered for deletion first increasing the chance of border cases being retained but also of noise retention. To combat this increased sensitivity to noise RT3 introduces a noise filtering pre-processing stage similar to the ENN algorithm: a case is removed if it is misclassified by its k nearest neighbours. Wilson & Martinez's (2000) DROP1-5 algorithms incorporate RT1-3 with slight extensions in DROP4 and DROP5. DROP4 has a less aggressive noise removal stage than RT3 and DROP5 considers cases nearest to an *enemy* for deletion first resulting in smoother decision boundaries than RT2.

Brighton & Mellish (2001) use a similar approach with their hybrid Iterative Case Filtering Algorithm (ICF) that combines a noise reduction pre-processing stage with a backward elimination compaction approach. ENN is used to pre-process the data and remove noise. Using coverage and reachable sets, modified for use in classification tasks, a case is deleted if its reachable set is larger than its coverage set, i.e. more cases can solve the case than it can solve itself. The process is repeated until

no more cases are being removed. This results in border cases being retained and central cases being removed.

Smyth & Keane's (1995) Footprint Deletion Policy is a heuristic approach based on their competence model. Cases are classified based on the contents of their reachability and coverage sets. Cases are then deleted from the case base in the order of their classification: auxiliary, support, spanning and then finally pivotal cases. Further sub strategies are then used to determine the order of case deletion within each category. This is one of the few approaches in which the final size of the edited case base can be controlled. However, it is an intuitive approach that does not guarantee competence will be preserved. Figure 2.3 below shows 5 cases. C1 is a pivotal case, C2 is a spanning case and C3 to C5 are auxiliary cases. If this group were reduced to one case by the footprint deletion policy C1 would be retained giving 2/5th the original coverage whereas if C2 were retained 4/5th the original coverage would be retained. Hence, while the approach is not ideal for case base editing the concept of classifying cases is interesting for competence modelling.

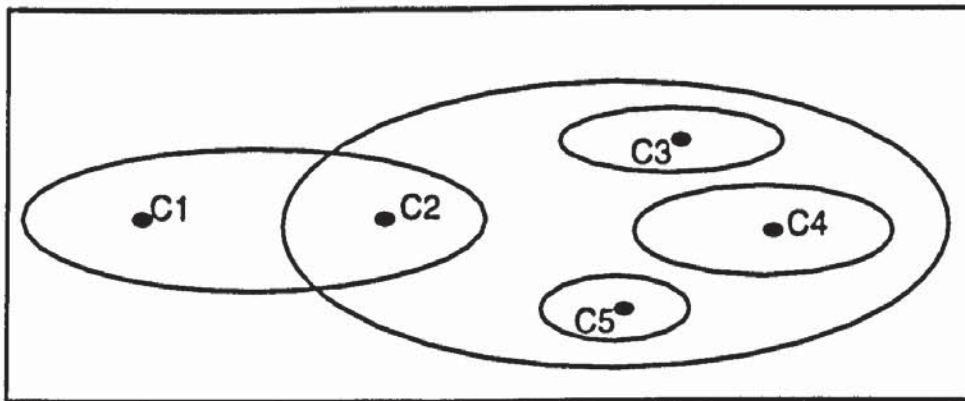


Figure 2.3: Case Coverage and Classification

The backward elimination algorithms provide an effective approach to compaction but are computationally demanding. In particular, the more modern hybrid approaches, RT3 and ICF, overcome the problem of retaining noise and provide a significant compaction of most case bases. They manage to achieve this while preserving competence by using local case competence knowledge to retain border cases.

Generalisation Compaction Algorithms

The algorithms in this section offer an alternative approach to removing cases and aim to reduce storage requirements and improve efficiency by modifying the cases themselves rather than by simply selecting which ones to keep. Chang's (1974) algorithm repeatedly tries to merge the nearest two cases of the same class into a single new case. This results in a case base that contains constructed cases as well as actual cases. The constructed cases are called prototypes. The merging process continues until classification accuracy starts to suffer. Prototypes are useful in characterising a class but poor at discriminating between classes because border cases are removed.

Domingos's (1995) RISE 2.0 system treats each case as a rule. For each rule the nearest case not covered by it is found and the rule generalised to include the new case, if accuracy is not affected. This process continues until no more rules can be generalised. The nearest rule to a new problem's input vector is used to provide its classification.

Salzberg's (1991) EACH algorithm used nested generalized exemplar theory in which hyper-rectangles are used to take the place of cases thus reducing storage requirements. Initially hyper-rectangles are created round randomly selected cases and then incrementally allowed to grow by looking at the nearest case and expanding them to include the new case, if it has the same class. Wettschereck & Dietterich (1995) introduced a hybrid nearest-neighbour and nearest hyper-rectangle algorithm that uses a hyper-rectangle to classify a new problem if it falls inside one and k -NN to classify problems not covered by any hyper-rectangle. This algorithm does not reduce storage requirements as the entire case base must be retained but it does improve efficiency by using as few hyper-rectangles as possible.

Generalisation approaches can offer a considerable compaction of the case base giving improved efficiency. However, there may be a reduction in competence as central points tend to be represented and decision boundaries are smoothed. There has been limited recent CBR research in generalised approaches and few experimental evaluations exist to allow fair comparisons to be made between the performance of generalised and selective approaches.

Case Reduction Summary

This section has identified and discussed the main features by which case reduction algorithms can be differentiated. Considerable research has been conducted in this area resulting in a large number of effective case reduction algorithms. The most notable algorithms have been classified (see Figure 2.2) and reviewed.

Compaction algorithms require a trade off between the level of compaction and competence preservation. The more modern algorithms (RC-CNN, RT3 and ICF) all provide a good balance between these conflicting objectives. RT3 and ICF both achieve their performance by using a noise filter as their first stage before retaining border cases. Experiments show they have similar performance with neither algorithm being consistently better across a range of case bases. RC-CNN can provide a greater compaction of the case base by initially retaining prototypical central cases followed by misclassified border cases. In some domains where the decision boundaries are less well defined or where there are large numbers of small clusters it may outperform the others. RC-CNN does not use a noise filter but one can easily be used if the domain demands it.

There are not sufficient experimental results to fully evaluate the merits of each algorithm. All the modern algorithms have informed, non-deterministic stopping criteria and achieve differing levels of compaction. Thus even where experimental results do exist evaluating the algorithms is difficult because different stopping criteria make direct comparisons impossible. In any case, the choice of *best* algorithm will depend on the characteristics of the particular domain. No one algorithm can be called the best, rather the knowledge engineer of a CBR system must choose the most appropriate algorithm using his knowledge of the domain and the performance requirements of the system, in terms of efficiency and storage constraints.

2.2.2 Case Discovery

Case discovery is the process of discovering new cases to fill gaps in the case knowledge with the aim of improving system competence. It may be required for a number of reasons. When a CBR system is being developed, or at an early stage in its implementation, a common problem is a lack of knowledge in the case base, i.e. cases are scarce. All the

cases available will be included in the case base but gaps in the case knowledge will reduce the system's competence. In established CBR systems internal or external changes in the environment, the task being undertaken or the user of the system can occur. These will affect the relationship between knowledge held in the case base and that required to complete the current problem-solving task.

In both these situations it is useful to identify gaps in the case knowledge and generate new cases to fill them. This process, called case discovery, is generally undertaken by a domain expert but research has been looking for ways to either assist the expert or develop automated techniques.

Case discovery is a different and more complex problem compared to the case reduction problem looked at in the previous section. In case reduction, the case knowledge is known beforehand making it easier to identify harmful or redundant cases within the problem space. In contrast, the task of case discovery is to add to the case knowledge, by using implicit information within the case base to identify areas within the problem space that when filled with a case will improve system competence. This is a difficult task and limited research has been published in this area.

The case discovery problem can be split into two tasks. First there is the need to discover *important* areas or holes within the case base and secondly there is the need to suggest or create cases to fill these gaps. The hole discovery and case creation task will be looked at separately.

Hole Discovery

One approach to identifying gaps (Liu, Ku & Hsu 1997) has been to focus on locating maximal empty hyper-rectangles (MHR) within k-dimensional continuous space and rank them according to their size. The algorithm starts with one MHR occupying the entire problem space. Each case, in the existing case base, is then added incrementally with the set of MHR's being updated at each insertion. When a new case is added all the MHR's containing this case are identified and removed from the set of MHR's. Using the new case as reference, a new lower and upper bound for each dimension are formed to result in two new hyper-rectangles along each dimension. If these new hyper-rectangles are sufficiently large, they are inserted into the set of existing MHR's, otherwise they are

rejected. This approach is limited to continuous attribute values and is computationally demanding. In an extension of this approach Liu, Wang, Mun & Qi (1998) develop an algorithm capable of locating MHR's within data containing both continuous and discrete valued attributes. The restriction that the hyper-rectangle is empty is relaxed to allow the presence of outlying cases within the proposed hole. The algorithm requires a minimum volume to be specified above which the MHR is considered interesting. The permitted number of outliers per MHR must also be specified as a user parameter. However, the main problem with this approach is there is no way to identify if the gap found in the problem space is *interesting* from a problem-solving view-point, it may simply be an impossible combination of attribute values.

McSherry's (2001) CaseMaker system uses a case discovery technique developed from its coverage model. A complete set of all uncovered cases is found by searching the space of allowable feature combinations. These cases are ranked based on their potential coverage contributions calculated using a linear adaptation function based on feature differences. This approach suffers from similar drawbacks as the coverage model. It only applies to nominal valued attributes, domain knowledge is required to identify valid attribute-value combinations, a linear adaptation function based on feature differences can be limiting and the ranking of proposed cases does not reflect the type of problems that will be encountered.

McKenna & Smyth's (2001a) approach to hole discovery, based on their competence model, is that interesting holes are those in the space between competence groups. In addition, the space between *nearest neighbour* competence groups is more likely to contain interesting holes than competence groups far from each other. The rationale for this is that they are more likely to reflect the absence of a valid case in an active region of the problem space. For each pair of competence groups a pair of boundary cases are identified. These boundary cases are identified as the closest cases (one case from each group) using the similarity metric. Next the nearest neighbour competence group is identified as the one containing the most similar boundary case. A hole exists between each competence group and its nearest neighbour. The boundary cases identify the extreme points of the hole. The holes are then ranked with the most similar boundary case pair identifying the most interesting hole. This is an intuitive approach, that assumes interesting holes are

located between competence groups, rather than a choice based on optimising a competence metric. Holes between competence groups may be interesting, particularly during the early evolution of a case base, however there is no guarantee that the space identified represents the *most interesting* hole or even a valid case, as it ignores large parts of the problem space and no consideration is given to domain constraints.

Case Creation

It is typically left to the domain expert to create a case to fill the hole in the problem space. However McKenna & Smyth (2001a) suggest an approach, as a second stage of their hole discovery algorithm, by trying to create a spanning case between the boundary cases to merge the competence groups. Given a pair of boundary cases, representing the extremes of the hole, they propose creating a case that lies between them. The feature values of the new case are set as the mean values of the cases in the *related sets* of the boundary pair. The related sets, calculated using their competence model, are union of a case's coverage and reachability sets. If the attribute values are discrete a majority vote is used to find the new case's feature value. Smyth and McKenna accept that human intervention would generally be required to validate the created case.

Case Discovery Summary

In contrast to the large research effort on case reduction techniques there has been only a limited effort applied to solving the case discovery problem. The problem is certainly not solved and much work remains to be done. Liu et al. (1998) identify holes in data but provide no measure of whether they are interesting, McSherry's (2001) approach applies to only a very few CBR domains and McKenna & Smyth (2001a) propose an intuitive approach that ignores large areas of the problem space. There has been very limited experimental results to support any of these approaches.

2.3 Case Visualisation

Visualisation tools have been used most frequently in CBR to present the system's proposed solutions. However, as CBR systems are becoming more complex, the process of

authoring and maintaining the case knowledge is becoming more difficult. Visualization techniques have the potential to improve this process by presenting the case base as a whole in terms of the relationship between cases or groups of cases (Smyth, Mullins & McKenna 2000). This allows users to better understand the structure of the evolving case base and may identify areas of the case base that are over or under populated, allowing the author to concentrate on these regions when adding or deleting cases.

We are interested in visualisation of the case base in order to assist the user understand the competence model being developed and implement the case discovery and deletion algorithms. The main problem in visualising the case base is to display in two or perhaps three dimensions, possibly a large amount of, multi-dimensional data . This section looks at three methods that have been used in CBR to visualise case bases: scatter graphs, force-directed graphs and parallel co-ordinate plots.

2.3.1 Scatter Graphs

Scatter graphs are X-Y plots with data indicated by symbols and are generally used to investigate the relationship between two variables. Weka (Witten & Frank 2000), an open source collection of machine learning algorithms for data mining tasks developed at The University of Waikato, New Zealand, uses scatter graph visualisation. In this approach, any of the attributes used in the case representation can be allocated to the X and Y axis. This allows the relationship between two attributes to be studied but is of little use for complex cases with many attributes or where attributes have nominal values.

McKenna & Smyth (2001b) developed a visualisation of their competence model as part of the CASCADE authoring system. The scatter graph displays competence groups plotted by the log of the group coverage (X-axis) and the log of the group size (Y-axis). Groups are connected to their nearest neighbour group by an arc to assist the case author judge the relationship between groups. A group can be selected to show its constituent cases. This tool enables an author to improve the competence of a case base by providing visual feedback to identify large competence groups containing redundant cases and small, low coverage competence groups which may require additional cases. The visualisation is dynamic and groups will merge or split as cases are added or deleted. However the visualisation gives little information about the underlying data, the relationship between

cases or the similarity knowledge being used.

Scatter graphs provide a simple, easily understood visualisation that allow relationships between individual attributes to be studied. However they are not good at displaying multi-dimensional data and give little insight to the structure of a case base or the similarity knowledge being used in retrieval.

2.3.2 Force-Directed Graphs

Undirected graph drawing approaches have been used to visualise the case base in CBR. In graph drawing, objects are placed on a page as nodes and the relationship between the objects are shown as edges. An undirected graph is one in which the nodes are connected by undirected edges, i.e. the edge has no directional arrow and both ends are equivalent.

Eades (1984) introduced the spring-embedder model which aims to provide aesthetically appealing graph layouts based on the following criteria: uniform edge lengths and symmetry as far as possible. Nodes in the graph are replaced by steel rings and each edge is replaced by a spring. The spring is associated with attraction and repulsive forces, according to the connecting distance between the nodes. The spring system starts with a random initial layout and the nodes are moved under the spring forces. An optimal layout is achieved as the energy of the system is reduced to minimum.

Kamada & Kawai (1989) introduced an extension to Eades spring-embedder model in which nodes are moved into new positions one at a time so that the total energy of the spring system is reduced with the new layout. The concept of a desirable distance between two nodes is also introduced, the distances can be set according to the requirements of the graph. This is a simple, intuitive approach but has been shown to work well. Kamada and Kawai's algorithm has been extended from a two-dimensional space to a three-dimensional version.

Mullins & Smyth (2001) used the spring-embedder model to develop a visualisation tool that aims to preserve the similarity relationships between cases as on-screen distances. The zero energy edge length, which represents the relationship between two cases, is set to be inversely proportional to the similarity between cases, using the similarity metric. The algorithm uses the attraction and repulsion of the *springs* to spread the cases around a two dimensional graph in an attempt to preserve the n-dimensional distances between cases.

Experiments on one case base show a correlation between the actual distance between cases and their screen distance of greater than 70%.

McArdle & Wilson (2003) use a similar spring-based algorithm, in which the edge length is set to be proportional to the distance between cases, to develop a dynamic visualisation of case base usage. Colour gradients are applied to cases in the visualisation in order to represent usage frequency, age of the case, or time since the case was last used for classification. This visualisation aims to assist manual case base maintenance by supporting the maintenance of large case bases.

These force-directed approaches provide more insight into the similarity assessment than the usual single dimensional value. However, the knowledge held within the similarity metric is hidden and the underlying data is not available. In addition case positions are not static, adding a new case can result in a complete rearrangement of the layout and successive runs of the same case base can result in a different layout.

2.3.3 Parallel Co-ordinate Plots

An alternative approach to visualising multi-dimensional data, originally proposed and implemented by Inselberg (1985), is the parallel co-ordinate plot. A parallel co-ordinate graph's primary advantage over other types of statistical graphics is its ability to display a multi-dimensional vector or case in two dimensions. Each attribute is represented by a labelled vertical axis. The value of the attribute for each case is plotted along each vertical axis. The points are then connected horizontally using line segments such that each case is represented as an unbroken series of line segments which intersect the vertical axes. Each axis is scaled to a different attribute. The result is a *signature* across n dimensions for each case. Cases with similar data values across all features will share similar signatures. Clusters of like cases can thus be discerned, and associations among features can also be visualised.

Falkman's (2002) Cube uses this approach to develop an information visualisation tool which displays a case base using a three dimensional parallel co-ordinate plot. The third Z-axis is typically used as a time line but can be used for any attribute. The arrangement of the axes can be important in parallel co-ordinate plots. The Cube uses an axes arrangement developed by Ankerst, Berchtold & Keim (1998), based on the similarity

between axes, in order to reduce line crossing on the graph. A similar approach is exploited in FormuCaseViz (Massie, Craw & Wiratunga 2004a, Massie, Craw & Wiratunga 2004b) to aid explanation in a pharmaceutical tablet formulation application. The problem and solution component of a case are displayed in separate parallel co-ordinate plots. The visualisation provides an explanation of the system's proposed solution by displaying cases in the neighbourhood of the target problem and allowing similarities and differences to be viewed at an attribute value level.

Parallel co-ordinate plots allow the underlying data to be visualised and can be useful for finding patterns or correlations within the data. However they can become very cluttered when large amounts of cases are being viewed and it is difficult to follow the signature of an individual case. While differences between the individual attribute values of cases can be viewed, it is difficult to assess the overall similarity between cases as the similarity values are not shown and knowledge held within the similarity metric is still hidden.

2.3.4 Visualisation Summary

In this research, visualisation is primarily required to display gaps and redundancy within the case base in a form that is useful to the knowledge maintenance engineer. In order to fulfil this requirement the visualisation will need to display information on the structure of the case base, the relationship between cases and some of the knowledge gained from competence modelling and the maintenance algorithms being developed. None of the visualisations looked at so far provide the full spectrum of information required, rather they each provide part of it.

McKenna & Smyth (2001b) make good use of clustering to provide competence modelling information in their scatter graph approach. However, they do not adequately represent the relationship between individual cases or clusters of cases. McArdle & Wilson (2003) display the case base structure and the relationship between cases in terms of the similarity metric in their force-directed graph but fail to provide clustering or competence modelling information. Parallel co-ordinate plots can provide the relationship between cases at a detailed, attribute level but have difficulty in displaying the general structure of the case base.

On a final point it should be remembered that visualisation of a case base becomes increasingly more difficult as its size increases. This is because it becomes more difficult to project multi-dimensional relationships onto 2-dimensions and many of the algorithms have difficulty handling large datasets. Clustering and indexing are techniques that can be used to address this problem.

Chapter 3


Problems with Existing Case Base Models

Case-Based Reasoning is often adopted as the problem-solving approach when domain knowledge is incomplete and, as a result, the relationship between individual cases and the problem-solving ability of the system is unknown. However, in many CBR systems, decisions still need to be taken about cases in the case base.

Imagine a knowledge engineer given a case base at the initial system design stage. Figure 3.1 shows a snippet of a example case base of an email dataset for classification between spam or non-spam. It is a flat attribute/value representation with the attributes along the top being selected stemmed words and the value representing the presence or absence of the word. There may be 1000's of cases and 100's of attributes. Given this data, the knowledge engineer designing a case base system has no idea if:-

- an easy or difficult problem is being faced.
- there is a shortage of data or too much with many redundant cases.
- the data contains erroneous cases, i.e. is it noisy?

A competence model of a case base will provide a simplification of the real system that can assist the knowledge engineer to make informed decisions about the case base. In CBR, modelling a case base can be considered to be the process of representing the case base in such a way that it gives a global view of the structure of the case base with



class	linguist	language	free	univers	.
SPM	0		0	0	.
SPM	1		1	0	.
SPM	0		1	0	.
NSPM	1		0	1	.
NSPM	1		0	0	.
.
.
.
.

Figure 3.1: Typical flat attribute/value representation of a case base

some overall predictive power, while at the same time providing local case information on individual cases or groups of cases that allow informed case base maintenance policies to be developed.

Globally, it is possible to measure how well a case base model estimates overall accuracy by comparing the model predictions with accuracy estimates obtained using leave-one-out testing, cross validation or training/test set approaches. In contrast, it is difficult to evaluate the local information supplied by a model directly as there is no comparable experimental results with which to compare the information. However, evaluation of the local information can be made indirectly by applying the local case information for particular maintenance tasks and comparing the results of the maintenance against alternative approaches. Hence, the first stage in evaluating the usefulness of a model is to measure how well the model estimates accuracy while the second, and more difficult stage is to apply the local information within particular case maintenance tasks and evaluate the results.

In this chapter McKenna & Smyth's (1998) existing competence model is reviewed and its performance on classification tasks is evaluated. In Section 3.2 we look at the disjoint nature of the problem space in classification problems and consider how this affects problem-solving accuracy and system competence.

3.1 Evaluation of Existing Case base Models

Competence is a measure of a CBR system's ability to perform its primary task. As CBR is a problem-solving methodology, competence is usually taken to be the proportion of problems faced that it can solve successfully. However, the actual problems that a system will face are unknown in advance and there are generally far too many potential target problems to consider all of them. Typically the case base is considered to be a representative sample of target problems and then, competence can be approximated by either test set accuracy or cross-validation experiments using the case base as a source of data.

Smyth & McKenna (1998) developed a competence model of the case base that measures coverage. The model is based on two assumptions; first that the case base is a representative sample of target problems and second that similar problems have similar solutions. The model groups together cases that solve each other in a four stage process. First, leave-one-out testing is used to give a measure of the problem-solving ability of a case using two important notions: coverage and reachability (Smyth & Keane 1995). Coverage of a case is the set of problems that the case can solve; conversely, reachability is the set of all cases that can solve it. Next, clusters of cases are formed using their reachability and coverage sets to group cases that have overlapping sets. These are called competence groups and form mutually exclusive sets of cases. The coverage of each competence group is then defined to be directly proportional to the size of the group and inversely proportional to the group's density as given by the formula below:

$$GroupCoverage(G) = 1 + [|G| \cdot (1 - GroupDensity(G))]$$

The $GroupDensity(G)$ of a group of cases G is the average case density of the individual cases within the group; where the case density of a case is the average normalised similarity to the other cases in the group as defined below:

$$CaseDensity(c, G) = \frac{\sum_{c' \in G - c} Sim(c, c')}{|G| - 1}$$

where $Sim(c, c')$ is a value between 0 and 1 calculated using the similarity measure defined for the domain applied to cases c and c' .

In the final step the overall coverage of the case base is simply the sum of the coverage of each group.

3.1.1 Evaluation

The true test of Smyth & McKenna's (1998) competence model is whether it reliably predicts the problem-solving ability of a CBR system. Their experimental results are reported for a recommender system using the package holiday and property datasets from the AI-CBR¹ web site. In these experiments coverage is shown to correlate closely with test set accuracy (McKenna & Smyth 1998, Smyth & McKenna 1998).

In classification tasks one of the assumptions made by Smyth & McKenna does not always apply. Similar problems do not necessarily have similar solutions. There are many areas of the problem space that will have the same solution but also areas, at class boundaries, that will have different solutions. Even although the discontinuous nature of the problem space in classification tasks violates one of the assumptions on which the model is based, the local information provided by the model has still been used on maintenance tasks in classification problems (Brighton & Mellish 2002, Delany & Cunningham 2004). We wanted to evaluate the ability of the model to predict accuracy in classification tasks.

We carried out experiments using Smyth & McKenna's model applied to classification problems. For the classification scenario, we adopted the same convention as Brighton & Mellish's (2001) to identify the reachability and coverage sets. The reachability set of a case is deemed to be the case's nearest neighbours that belong to the same class (i.e. the most similar k cases retrieved by the k Nearest Neighbour algorithm (k-NN) which predict the same class) but bounded by the first case belonging to different class. The idea behind this approach is that competence groups should not jump across areas of the problem space containing cases belonging to a different class. Using this approach, the competence model was implemented on four classification datasets from the UCI Machine Learning Repository (Blake, Keogh & Merz 1998): iris, house-votes, tic-tac-toe and zoo.

¹<http://www.ai-cbr.org>

Method

Each of the datasets was randomly split into a training set and test set approximately in the ratio 70:30. The training set provides cases for the case bases and the test set provides a collection of unseen target problems. The training sets contain 50, 70, 250, and 700 cases for zoo, iris, house-votes and tic-tac-toe respectively. Initially each training set was partitioned into n disjoint subsets. The smallest case base was created using one of these subsets, and a growing case base was created by successively adding either one or more of these subsets. After all the cases in the training set had been added, larger case bases were formed by adding duplicate cases to introduce redundancy into the case base. The introduction of redundant cases allows us to evaluate the model on case bases containing cases that contribute little to competence and where the number of cases is less important.

Test set accuracy was calculated, for each case base, by using a standard 3-NN retrieval and weighted majority vote to predict the class of the unseen problems and comparing this prediction with the problem's actual class. The competence model was applied to each case base and its coverage calculated and compared to the test set accuracy.

Results

Figure 3.2 shows the average results from 20 experiments on each case base size on each of the four datasets. The solid line on each graph shows the accuracy obtained when using the unseen test set to evaluate the different sized training case bases. Coverage calculated by the model has been normalised between 0 and the maximum accuracy obtained on that dataset. The dashed line on each graph shows the competence prediction for the different sized case bases.

As expected, test set accuracy initially increases as the case base size grows and then peaks at a maximum value when a sufficient number of cases have been added to solve most of the new problems that are encountered in the test set. This levelling out in accuracy would be particularly expected as the redundant cases are added in the larger case base sizes. This behaviour is seen on the test set accuracy plots for the four datasets. However, on the coverage plots, coverage continues to rise even when the duplicate redundant cases are being added in the larger case base sizes.

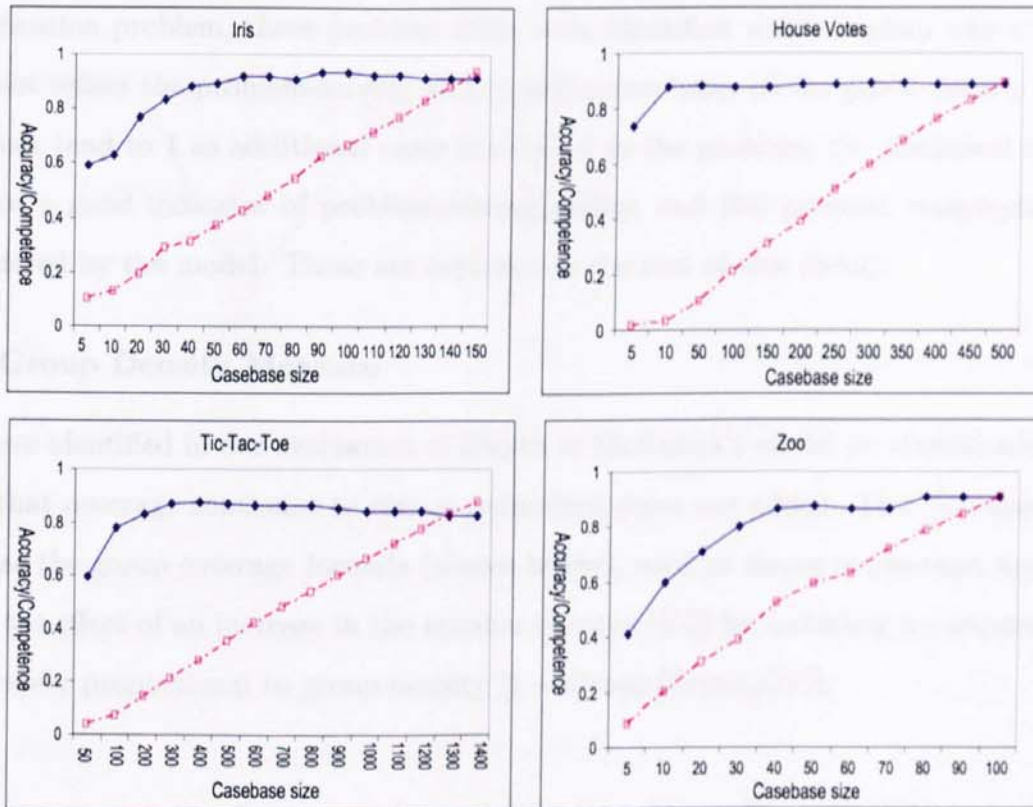


Figure 3.2: Graph showing accuracy on test set and competence prediction for different case base sizes on four datasets

It can be seen from these graphs that there is limited correlation for these classification tasks between the model's predictions, given by coverage, and test set accuracy.

3.1.2 Discussion

Smyth & McKenna's (1998) model incorporating the Brighton & Mellish (2002) extension does not appear to produce an accurate reflection of problem-solving ability on classification problems. The coverage given by the model appears to have a strong relationship to the number of cases in the case base. It may correlate quite closely with competence for smaller, sparse case bases where adding to the number of cases gives a corresponding increase in a system's accuracy. However, there does not seem to be any correlation for larger case bases containing duplicate redundant cases in which adding new cases is not expected to give a corresponding increase in accuracy.

We wanted to explain the low correlation between test set accuracy and the model's

prediction. Through visualising a range of problem-solving scenarios on a simple binary classification problem, three problem areas were identified which explain why the model may not reflect the problem-solving ability of the case base; (i) the group density measure does not tend to 1 as additional cases are added to the problem, (ii) statistical measures are not a good indicator of problem-solving ability, and (iii) problem complexity is not considered by the model. These are explored in the rest of this section.

The Group Density Measure

We have identified in our evaluation of Smyth & McKenna's model on classification problems that coverage continues to rise as redundant cases are added. This was unexpected because the group coverage formula (shown below), used to measure coverage, appears to offset the effect of an increase in the number of cases ($|G|$) by including a component that is inversely proportional to group density ($1 - \text{GroupDensity}(G)$).

$$\text{GroupCoverage}(G) = 1 + [|G| \cdot (1 - \text{GroupDensity}(G))]$$

On the face of it, this seems reasonable: as the number of cases in a group increases there will be a corresponding increase in group density. This should result in a group coverage that reaches a maximum and does not continue to rise as redundant cases are added. However, this is not what happens because the case density measure used (average similarity distance to the other cases in the group) does not tend to 1 as the number of cases increases toward infinity. A simple example is described now which demonstrates the group density of a competence group tending to an upper limit of less than 1.

With the density formula used here the actual maximum case density is dependent upon the composition of case similarities. If we look at the competence group represented in Figure 3.3(a), there are n cases spread evenly between three positions. The similarity between the different positions within the group is 0.5 and each position contains a third of the cases. Figure 3.3(b) shows a graph of the average group density calculated as specified in the model. It can be seen that the maximum density value reached is 0.67.

The effect of this formula, in which the number of cases can continue to increase while group density does not, results in a group coverage overly dependent on the number of

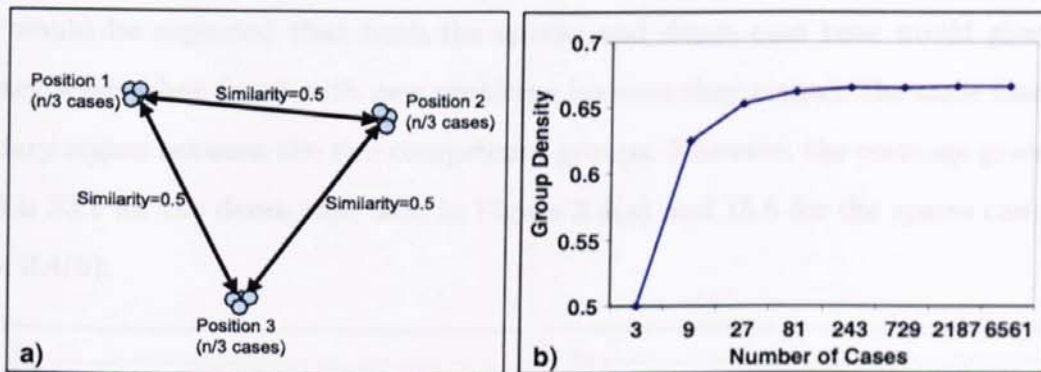


Figure 3.3: Relationship between case density and number of cases

cases in the group. This explains the results obtained in our evaluation in which group coverage continues to increase as redundant cases are added. This problem could perhaps be addressed by the use of an alternative case density measure that tends to 1, as the number of cases tends to infinity. However, the remaining two problems would still restrict the applicability of the model to classification tasks.

Statistical Measures versus Case Positioning

Smyth & McKenna's (1998) model uses the retrieval and adaptation mechanism of the CBR system to form the case base into clusters and thereafter statistical measures are used to determine the coverage of each cluster. The two statistical measures used are the number and density of cases in the cluster. The use of purely statistical measures to calculate coverage is a fundamental problem with Smyth & McKenna's model when applied to classification tasks.

To help demonstrate this problem we set up a simple problem-solving scenario with a binary classification problem comprising two numeric features. This allows the problem space to be visualised in two dimensions corresponding to the two numeric features. In Figure 3.4 each case is represented by plotting a symbol on the graph according to the values of its two features. The two classes are distinguished by the shapes square and circle. The two competence groups are shaded differently in black or white.

A dense case base is shown in Figure 3.4(a) and the sparse case base in Figure 3.4(b) is a subset of the dense case base. The boundary between the two classifications in each Figure is the same and so an identical problem-solving domain is being viewed. The sparse

case base retains most of the cases near the class boundary.

It would be expected that both the sparse and dense case base would give similar accuracy levels when faced with new problems because they contain the same cases in the boundary region between the two competence groups. However, the coverage given by the model is 33.1 for the dense case base in Figure 3.4(a) and 15.6 for the sparse case base in Figure 3.4(b).

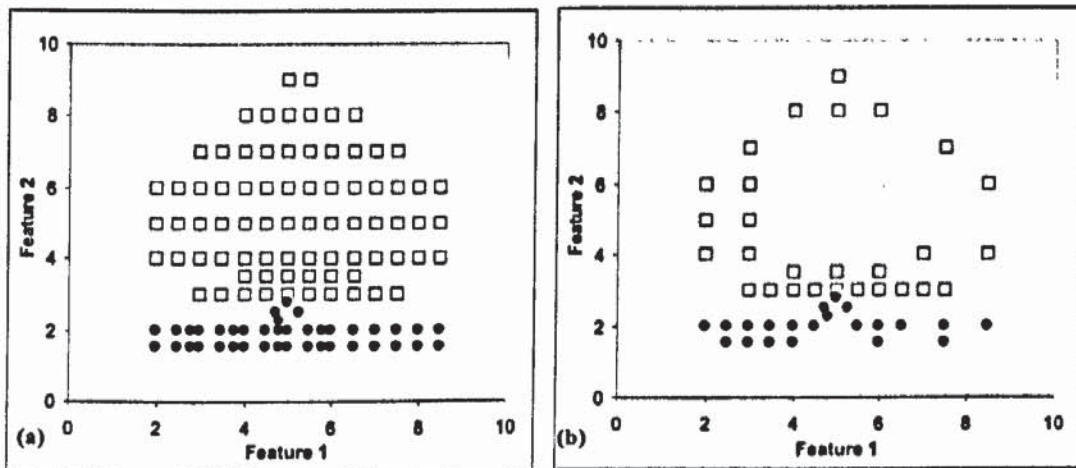


Figure 3.4: Visualisation of two different case bases for the same problem-solving domain

It can be seen from this simple example that statistical measures alone do not give a good prediction of competence because they ignore the position of a case. The position of a case is of crucial importance in classification problems because it is the boundary cases that mark out the transition from one class to another and determine the accuracy of a CBR system. For a model of classification problems to be able to predict competence it must measure the ability of a case to classify and this is, at least partially, dependent on a case's position in relation to a class boundary.

Problem Complexity

Modelling the competence of case bases can play an important role in benchmarking for CBR systems. Benchmarking can provide comparisons across problems and case bases or it can be limited to the comparison of different case bases within the same problem domain. Smyth & McKenna's model is restricted to benchmarking different case bases within the same domain. To highlight this limitation we use two similar problem-solving scenarios

with our binary classification problem and in this example each class corresponds to a single competence group. In these scenarios, shown in Figure 3.5, the boundaries between two classifications are different but the composition of cases within each group is the same.

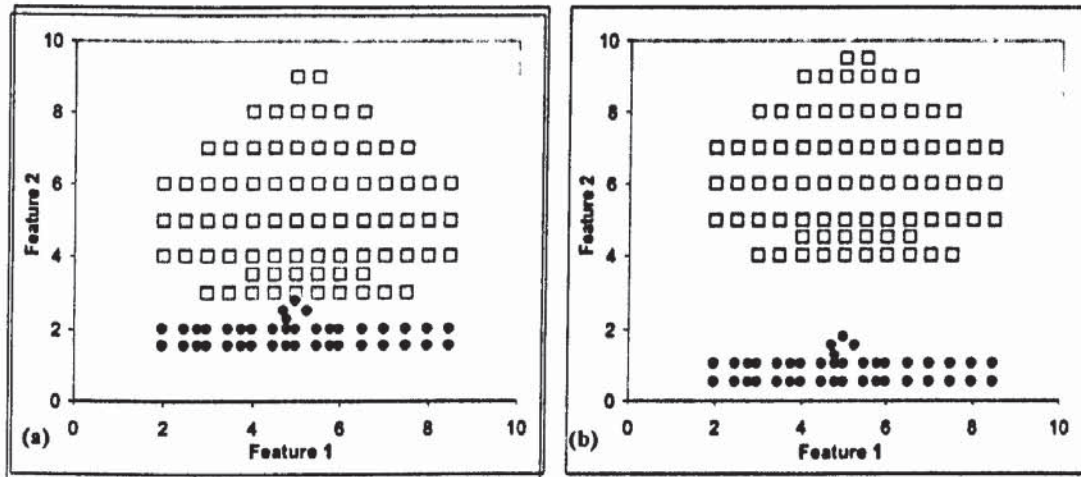


Figure 3.5: Visualisation of two problems with similar case base composition but different boundary conditions

In Figure 3.5(a) there is little separation between the classes at the boundary. The similarity between two cases with different classes can be high and there is a high probability that a CBR system will wrongly classify some problems in the boundary region. In Figure 3.5(b) there is a large separation between classes because there are no two cases with a different classification that have a high similarity value. In this situation, assuming the case base is representative of problems to be faced, it is less likely that new problems in the boundary region will be wrongly classified. However the predicted competence, measured by coverage, of the two situations will be identical because the composition of the competence groups is the same. The model does not appear to adequately reflect the complexity of the problem being faced in classification problems.

It can be seen from this simple example that Smyth & McKenna's model can not be used to provide benchmarks between different problem domains because it takes no account of the difficulty of the problem being faced. If a model is to provide benchmarks across problems it must consider problem complexity.

3.2 Issues with Classification Problems

In classification tasks similar problems do not necessarily have a similar solution. The disjoint nature of the problem space can result in cases with small changes in their feature values having totally different solutions. In the previous section we examined the Brighton & Mellish (2002) extension of Smyth & McKenna's (1998) competence model and found that it does not provide a close correlation with accuracy for classification problems. One reason for this is that the difficulty or complexity of a problem, introduced at the classification boundaries, is ignored.

In order to create a model for classification problems we must first try to understand the factors that affect the complexity of a problem. In other words what makes a classification problem *easy* or *hard*? The difficulty of a classification problem is affected by the cases available to represent the problem space but also by the inherent problem complexity contained within the problem space. In this section, we first look at inherent problem complexity and the availability of cases in more detail before investigating the impact of these factors on a typical accuracy graph.

3.2.1 Inherent Problem Complexity

Inherent problem complexity is a measure of how difficult the problem would be if the number of cases available in the case base was not an issue, or, in other words, the error rate of a problem with all possible cases available in the case base.

If we assume a problem to be represented by a fixed set of cases, each consisting of a collection of attribute/value pairs and an associated class label, then classification can be viewed as the process of partitioning the problem space into labelled regions. Decision boundaries are formed between these labelled regions.

A case is misclassified when it falls in a region which does not have the same class label as itself. Complexity can be measured by the proportion of samples that are misclassified; i.e. error rate. However, error rate is dependent on the classifier chosen in so much as different classifiers form different decision boundaries. We are looking for a measure that correlates well with a classifier's typical performance and provides localised information about areas within the problem space. In a situation where we have a complete sample

of all possible cases, a measure of the separability of the class labelled regions provides a good measure of the complexity of a classification problem.

A visualisation of two scenarios is shown in Figure 3.6. A case is represented by a symbol on the plot with the class of the case distinguished by the shape; star or circle. Class boundaries are formed by surrounding cases of each class with a boundary to form two labelled regions. In Figure 3.6(a) there is no overlap between the labelled regions and a margin of separation exists between the two classes. In Figure 3.6(b) there is an overlap region where the two labelled regions meet, shown as the shaded region. Overlap regions are the intersection of the class labelled regions and conceptually, misclassified cases will occur in these overlap regions.

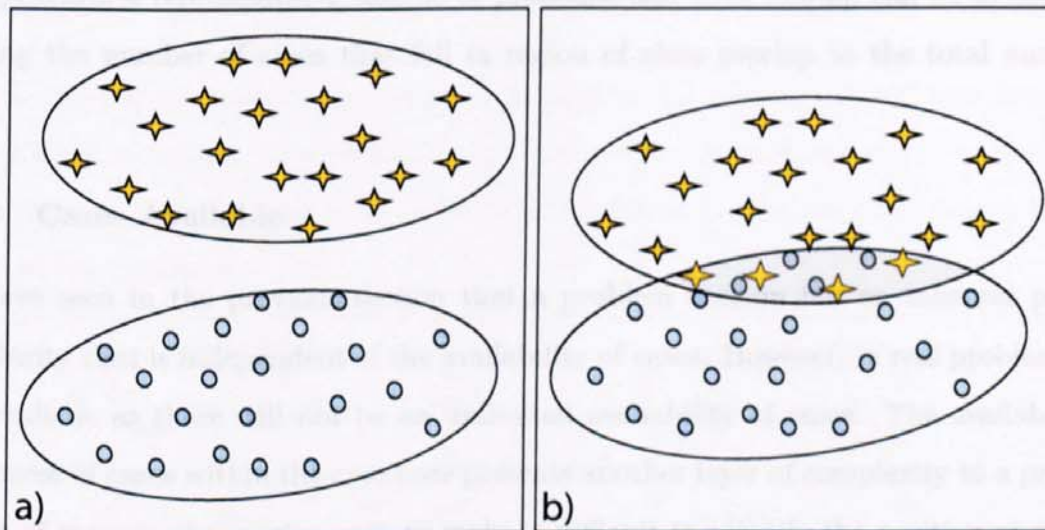


Figure 3.6: Effect of boundary overlap on accuracy in classification tasks

If the labelled regions are separable the problem can be considered *easy* and the larger the gap or margin between labelled regions the easier the problem becomes. Conversely, if the labelled regions overlap the problem becomes complex and the greater the overlap the more complex the problem becomes.

The inherent complexity of a classification problem is dependent on the size or volume of the overlap regions contained within the problem space. The size of each overlap region will depend on the length of the decision boundary and margin of overlap. The number of decision boundaries will depend, to some extent, upon the number of different classes that exist in the problem domain but more directly on the number of separate homogeneous

groups of cases that exist for each class.

In simple problems it may be possible to measure overlap directly, however, in multi-dimensional and possibly, multi-class problems with large or infinitely sized problem spaces measuring overlap directly becomes close to impossible. Measures that approximate overlap volume must be found and some of these will be discussed in the following sections.

Another concern is how to make comparisons between the level of overlap found, and the overall problem space. If problems that a system will face are equally distributed across the problem space then the volume of overlap in relation to the total size of the problem space would give a measure of inherent problem complexity. However, in real problems this is rarely, if ever, the situation. An alternative approach is to assume that the case base provides a representative sample of problems and then overlap can be measured by relating the number of cases that fall in region of class overlap to the total number of cases.

3.2.2 Cases Available

We have seen in the previous section that a problem domain has an inherent problem complexity that is independent of the availability of cases. However, in real problems, this is unrealistic as there will not be an unlimited availability of cases. The availability or sparseness of cases within the case base presents another layer of complexity to a problem. A lack of cases in the overlap regions make it difficult to identify the position of decision boundaries and can give a false impression of low complexity. However, in contrast, cases in the interior of labelled regions are irrelevant from a classification point of view and have no impact on problem complexity.

A visualisation of two scenarios, using our established conventions, is shown in Figure 3.7. These scenarios highlight that counting the number of cases or overall case densities is a poor indicator of problem complexity for classification problems. Both scenarios show an identical problem-solving domain with the same inherent problem complexity. Figure 3.7(a) contains dense labelled regions compared to Figure 3.7(b). However the sparse case base retains all of the cases near the decision boundary and would give a similar level of accuracy and hence level of complexity.

In classification problems, counting the number of cases is not a good indicator of the

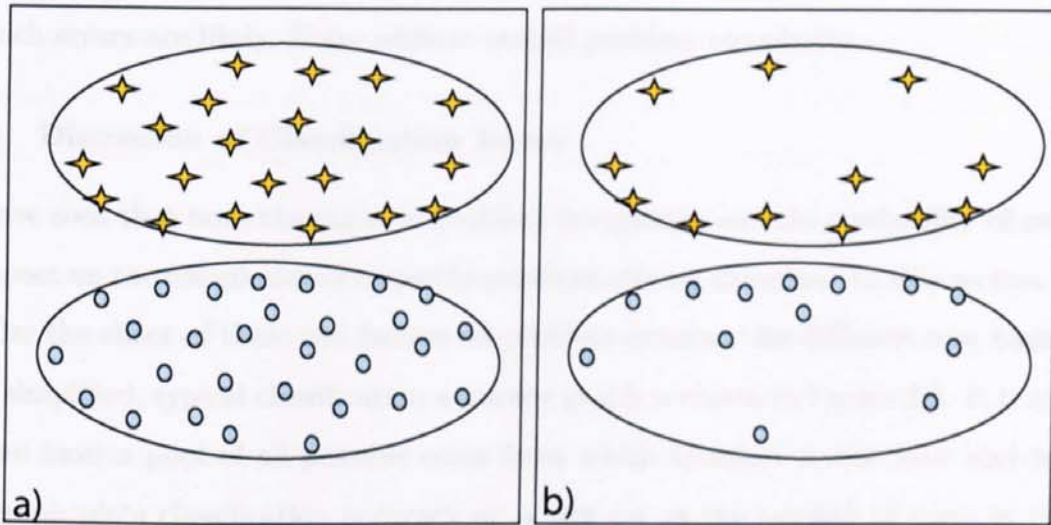


Figure 3.7: Effect of the number and position of cases on classification complexity

effect of the sparseness of cases on problem complexity. Likewise overall case density is not a good indicator. The positioning of cases within the problem space in relation to decision boundaries appears to be an important factor as to whether a case has an impact on problem complexity.

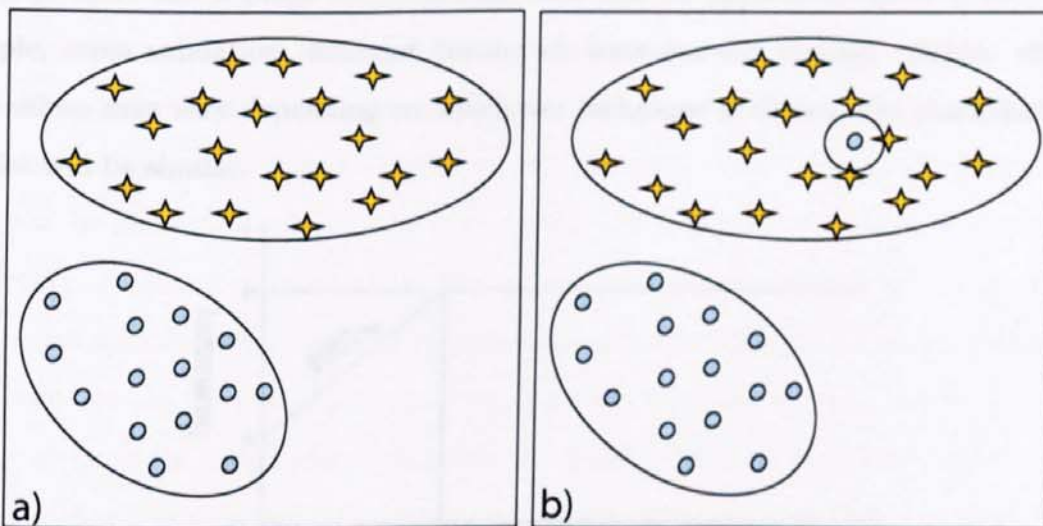


Figure 3.8: Effect of noise on problem complexity

A second important issue in relation to the availability of cases is the presence of erroneous or noisy cases in the case base. If we look at the two scenarios shown in Figure 3.8 the same problem domain is represented but Figure 3.8(b) contains a noisy, *circle* case located in the cluster of *star* cases. This introduces a false overlap region and

decision boundary into the problem space resulting in an increase in the number of regions in which errors are likely. Noise adds to overall problem complexity.

3.2.3 Discussion of Classification Issues

We have seen that both the inherent problem complexity and the availability of cases has an impact on the complexity of a specific problem-solving situation. In this section we will consider the effect of these two factors on problem accuracy for different case base sizes.

A simplified, typical classification accuracy graph is shown in Figure 3.9. It is assumed that we have a pool of all possible cases from which to select a case base and test set. The graph plots classification accuracy on a test set as the number of cases in the case base increases. There are a wide range of classification methods that can be used to classify the unseen problems and calculate accuracy, besides the k-NN algorithm used for classification experiments in this report, for example, support vector machines, naive bayes and C4.5 decision trees. While the decision boundaries and hence accuracy obtained by each classifier will be different, the graph obtained will have similar characteristics. Similarly, there are a range of techniques that can be applied to select a test set, for example, cross validation, hold-out testing or leave-one-out testing. Again, while the exact values may vary depending on whichever technique is chosen, the characteristics of the plot will be similar.

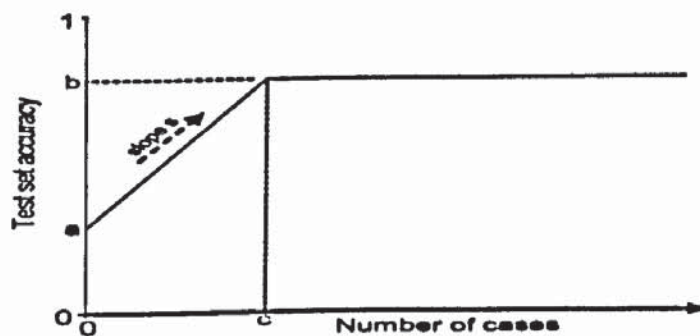


Figure 3.9: Typical graph of test set accuracy as a case base grows

The plot starts with its lowest accuracy where the plot intersects the y-axis. The accuracy then increases steadily with a slope 's' until the maximum accuracy of the system is reached. The accuracy typically then remains stable no matter how large the case base grows.

The minimum or starting accuracy, achieved without any case knowledge, is shown as 'a' on the graph. This corresponds to a guess as to the class of the unseen problems and is related to the number of classes in the problem but also on the distribution of cases between classes. The best starting accuracy that can be expected is the proportion of cases labelled as the class with most cases. The positioning of cases, obviously can not affect starting accuracy and, likewise, there is no relationship between inherent problem complexity and starting accuracy because no cases are available to define the decision boundaries.

The maximum accuracy achieved, shown by 'b' on the graph, after enough cases have been added to the case base to fully define the problem space. The maximum system accuracy is, by definition, fully determined by the inherent complexity of the problem. There is no relationship between the positioning of cases within the case base and the maximum accuracy achieved.

The slope of the increasing accuracy section of the plot is dependent on both inherent problem complexity and the positioning of cases within the case base. If a problem has low inherent complexity, perhaps because it contains few decision boundaries, then few cases are required to define the decision boundaries. As cases are added to the case base the decision boundaries will be quickly defined and accuracy will increase quickly resulting in a steeper slope. In a complex problem, with many or long decision boundaries, more cases will be required to define the boundaries and accuracy will increase slowly. The positioning of cases has an effect on the slope of the graph. If *discriminating* cases are chosen that quickly define decision boundaries, accuracy will increase more quickly than if the cases chosen are poor discriminators, possibly far from decision boundaries.

The slope of the graph also determines the number of cases required to achieve maximum accuracy identified by point 'c' on the graph. This could be considered an *optimal* size for the case base because it provides the quickest retrieval time to give maximum accuracy. Other operational factors may influence the actual size of case base used.

3.3 Chapter Summary

The evaluation of Smyth & McKenna's (1998) model on classification problems has identified datasets in which the model's predicted coverage does not give a good correlation to classification accuracy. Three reasons for this lack of correlation have been identified: the model does not adequately reflect the inclusion of redundant cases within the case base, statistical measures are not a good measure of competence in classification problems and problem complexity is not reflected by the model.

Local information gained from a model that more closely predicts competence on the global level could be expected to provide more useful information for case base maintenance policies. A model is required that focuses on the positioning of a case within the case base in relation to class boundaries. However, if we wish to make comparisons across problem domains the model must also be sensitive to the difficulty or complexity of the problem being faced.

In classification problems, the problem space can be split into regions where all cases have the same solution i.e. are labelled with the same class. These regions are separated by decision boundaries. The inherent problem complexity assumes that all possible examples are available. In this situation problem complexity becomes the complexity of the decision boundary and is dependent on the number and length of boundaries and the overlap at the boundary.

The availability of cases adds another layer of complexity to the problem. However standard statistical measures, such as a count of the number of cases or average case density, provide little help in identifying the effect of the availability of cases on problem complexity. Rather, the positioning of the cases in relation to the decision boundaries is the important factor.

The challenge in modelling the competence of classification problems is to quantify the size or effect of overlap regions in conjunction with the effect of the sparseness of cases available to the case base.

Chapter 4

Complexity Model for Classification Problems

We have seen that complexity is a characteristic of the problem-solving system, dependent on both the problem domain and the availability of cases, that gives a measure of how difficult it is to classify new problems. Complexity is determined largely by decision boundary conditions through such factors as the number, length and overlap of decision boundaries and the positioning of available cases in relation to the boundaries. Previous research on case base editing has also highlighted the importance of cases in boundary regions on the competence of a case base (Brighton & Mellish 2002, Wilson & Martinez 2000, Delany & Cunningham 2004). It seems reasonable to assume that in order to model competence, consideration must be given to the interaction between cases on boundaries.

Two modelling approaches have been investigated that consider interactions between cases at decision boundaries. We call the approaches the boundary approach and the complexity approach. In the Boundary approach, cases on decision boundaries are explicitly identified and these cases are used to calculate boundary measures that provide information about the decision boundary. In the Complexity approach, a complexity value is first calculated for each case by considering the class composition of the case's neighbours. A profile of these complexities is then plotted which provides the knowledge engineer with an insight into the structure of a case base from a global perspective.

In this chapter we introduce our two approaches to modelling the case base and show

how they both provide an indication of problem complexity and how the complexity approach, in particular, appears to provide a good correlation with test set error rates. In addition, the case complexity calculated as part of the complexity approach provides information on the level of uncertainty present in local areas of the case base that can be used to inform maintenance algorithms. We discuss the boundary approach in Section 4.1 and introduce our new complexity approach in Section 4.2.

4.1 Boundary Approach

We are attempting to create a model of a case base that will give an insight into the characteristics of the problem domain that the case base is supporting. We have identified that there is a relationship between complexity and the interactions between cases in regions of the problem space at or near decision boundaries. An obvious approach to modelling the case base is to attempt to measure these interactions directly. That is exactly what we attempt in this boundary approach. Cases on decision boundaries are explicitly identified and used to calculate measurements that aim to define the characteristics of a particular decision boundary.

4.1.1 Boundary Measures

This approach identifies cases near decision boundaries. Once these cases are identified, metrics that measure the distance between boundary cases, the length of boundaries and the density of cases on the boundary are applied. The boundary model is implemented in the following five stage approach:-

1. **Cluster the case base:** Clusters of cases, i.e. competence groups, are formed using the extension to Smyth and McKenna's competence approach for classification tasks described in Section 3.1.
2. **Identify cases on cluster boundaries:** Boundary cases for each cluster are found by looking to within the cluster from all the cases outside it. Figure 4.1 shows a representation of a case base with cases belonging to three different classes represented by the shapes; *circle*, *square* and *star*. In order to identify boundary cases we look

at each cluster in turn. If we consider the cluster containing circles first, then all the cases that do not belong to this cluster are selected, i.e. *star* and *square* cases. For each selected case their nearest neighbour case from within the cluster is identified, shown by the dark circle cases in Figure 4.1. These nearest neighbours are deemed to be the boundary cases for the circle cluster.

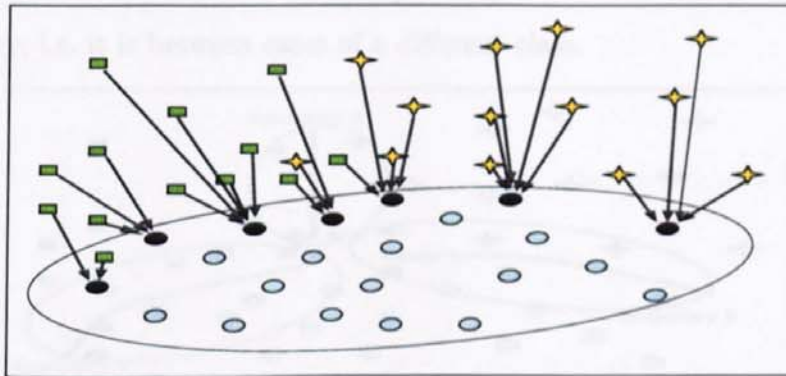


Figure 4.1: Identification of boundary cases within a cluster

3. **Identify individual cluster decision boundaries:** A cluster may have several boundaries; i.e. a boundary with more than one opposing group. These individual boundaries (i.e. with a single opposing cluster) are identified by looking from within the cluster to the cases outside it. The nearest neighbour from an opposing group is found for each boundary case. Figure 4.2 shows the boundary cases, related to the circle cluster, from the square and star clusters.

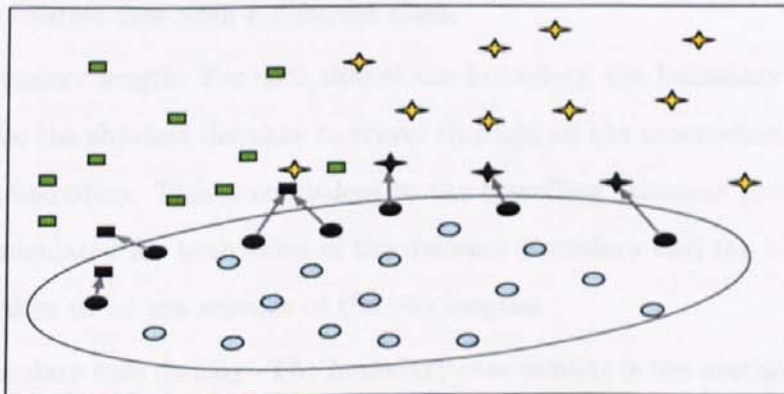


Figure 4.2: Identification of boundary cases from opposing group

4. **Identify decision boundaries:** After the previous two steps have been completed for each cluster, the individual boundaries are identified by grouping together boundary cases that have neighbours from the same opposing groups. Figure 4.3 shows the three decision boundaries for the example being considered, together with the cases selected to represent the boundary. Of course, cluster boundaries are not necessarily decision boundaries. The boundary is only considered further if it is a decision boundary; i.e. it is between cases of a different class.

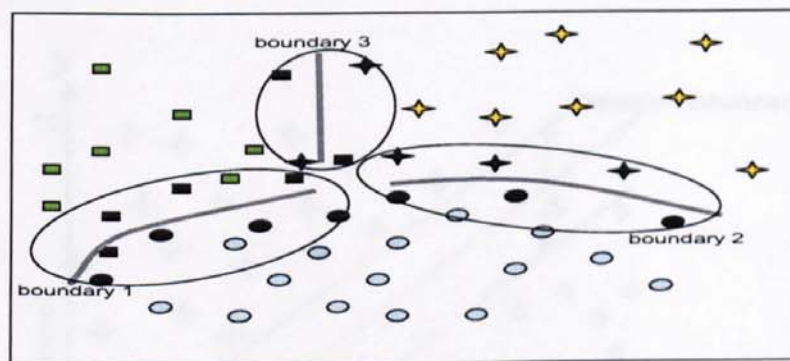


Figure 4.3: Identification of individual decision boundaries

5. **Calculate boundary metrics:** For each decision boundary, metrics giving information about the boundary conditions are calculated using the cases selected to represent the boundary and the similarity or distance function associated with the cases. The following three metrics are used:-

- **Boundary separation:** Separation at each decision boundary is calculated as the average distance from a boundary case to its nearest unlike neighbour; i.e. the nearest case with a different class.
- **Boundary length:** For each side of the boundary, the boundary length is taken to be the shortest distance to travel through all the cases selected to represent the boundary. This is equivalent to the travelling salesman problem. A length is calculated for both sides of the decision boundary and the boundary length is taken to be the average of the two lengths.
- **Boundary case density:** The boundary case density is the average density of the boundary cases. Where case density is the average similarity to its k -nearest neighbours.

4.1.2 Experimental Evaluation of Boundary Approach

One objective of our boundary approach is to create a model of the case base that will provide a global measure that correlates well with the complexity of a problem, measured by error rate or accuracy. In order to obtain this global measure the three boundary metrics (separation, length and density) need to be combined to provide a single measure. It is expected that accuracy will be directly related to separation and density but inversely related to length.

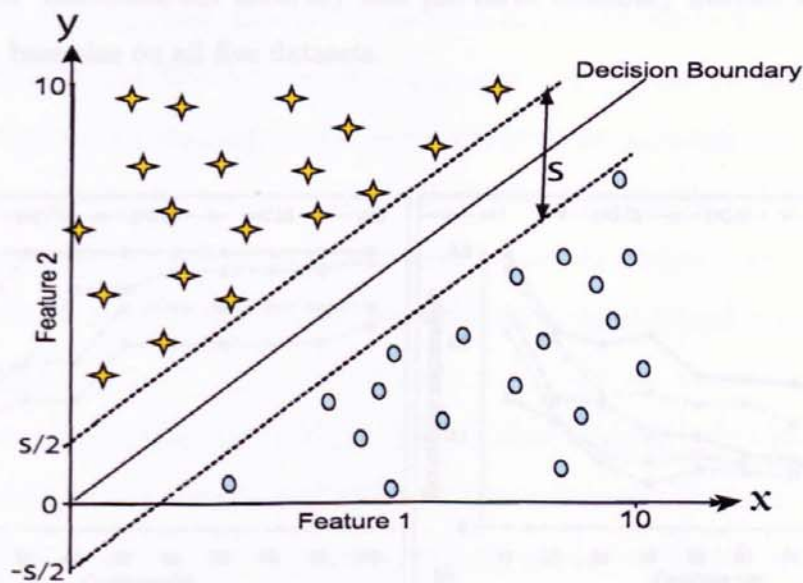


Figure 4.4: Graph of an artificial dataset identifying the decision boundary and separation between classes

To help investigate the characteristics of these boundary measures in relation to accuracy, we set-up a simple, artificial, problem-solving scenario with a binary classification problem comprising two numeric features (x and y) with values in the range 0 to 10. This representation allows the case base to be visualised in two dimensions corresponding to the numeric features. In Figure 4.4 each case is plotted by a symbol (star or circle) on the graph according to the value of its numeric features. A decision boundary at 45 degrees from the origin, representing the line $y = x$, is shown by the solid line. A case is created by first setting each feature to a randomly selected value between 0 and 10. Then the case is allocated the class star if it lies above the line $y = x + s/2$ on the graph, and circle if it lies below the line $y = x - s/2$; where s is the enforced minimum separation between the

classes, shown as the distance between the two dotted lines in Figure 4.4. A case is not accepted if it lies between the dotted lines. The complexity of the problem being faced can be changed by varying the enforced separation or the number of cases in the case base.

Experiments have been carried out on five similar datasets, each containing 100 cases. The enforced separation is different for each dataset, varying in steps of 0.25 between 0 and 1. Each dataset was split into 10 disjoint sets. The smallest case base was formed by selecting one of these sets and a growing case base was formed by successively adding an additional set. Leave-one-out accuracy and the three boundary metrics were calculated for each case base size on all five datasets.

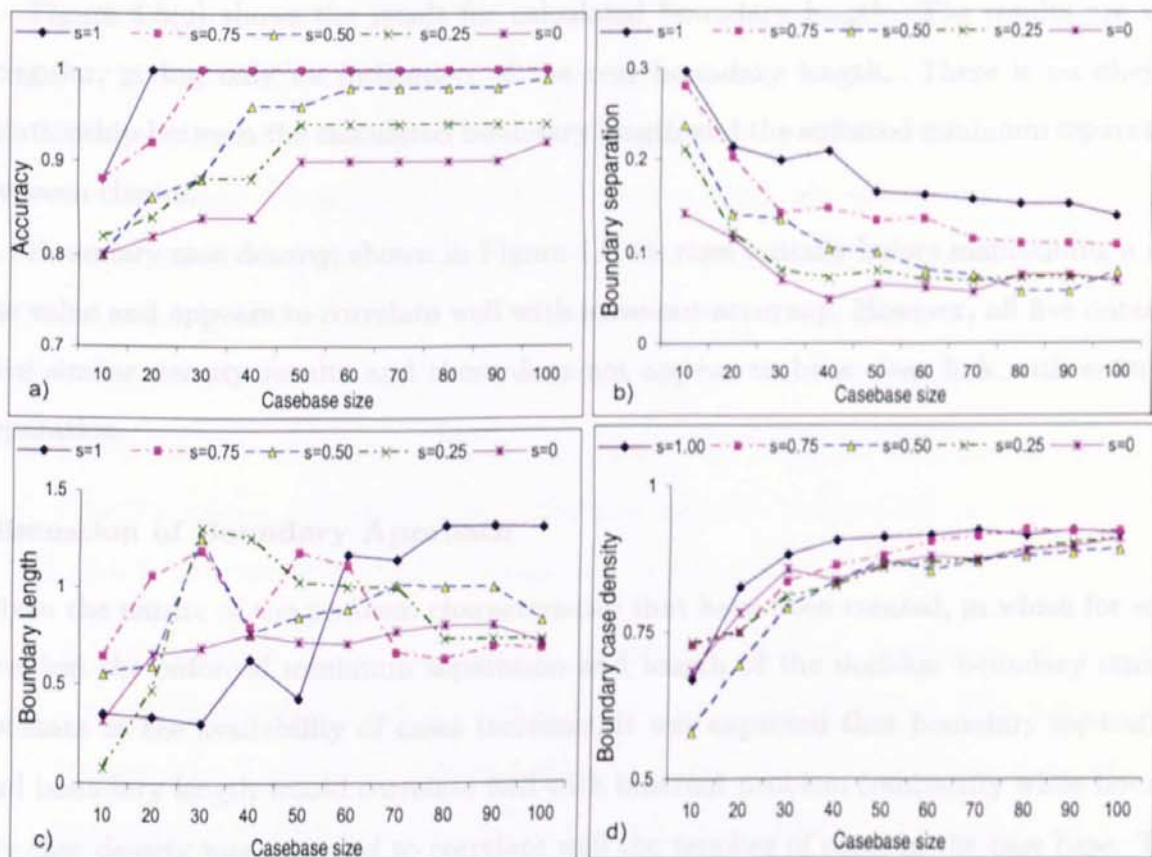


Figure 4.5: Graphs of leave-one-out accuracy and the three boundary metrics on an artificial dataset with varying boundary separation

The average results from 5 runs of the experiment on each dataset are shown in Figure 4.5. The results for leave-one-out accuracy testing (Figure 4.5(a)) show that for all five

datasets accuracy initially increases, as case base size grows, before reaching a maximum value. The greater the enforced separation between classes, the faster the accuracy grows and the maximum accuracy is achieved with a smaller case base size. In addition, the maximum accuracy is generally higher for problems with larger enforced separations although the case bases with enforced separations of 1.0 and 0.75 both reach 100% accuracy. These results confirm that reducing the separation between classes results in an increase in problem complexity.

The calculated boundary separation results (Figure 4.5(b)) show the boundary separation falling as the case base size grows and the boundary becomes more defined. The measure clearly identifies the differences in separation in the five datasets with plots for datasets with higher enforced separation lying above those with lower separation.

Figure 4.5(c) shows the result for calculated boundary length. The results are very irregular, giving only an indication of the real boundary length. There is no obvious relationship between the calculated boundary length and the enforced minimum separation between classes.

Boundary case density, shown in Figure 4.5(d), rises initially before maintaining a stable value and appears to correlate well with leave-out-accuracy. However, all five datasets give similar density results and there does not appear to be a clear link with enforced separation.

Discussion of Boundary Approach

Given the nature of the problem characteristics that have been created, in which for each problem the enforced minimum separation and length of the decision boundary remain constant as the availability of cases increases, it was expected that boundary separation and boundary length would correlate well with inherent problem complexity while boundary case density was expected to correlate well the number of cases in the case base. The results only partially support these expectations.

The boundary separation metric was expected to give a measure of the complexity of the decision boundary. The experimental results confirm that the metric is able to discriminate between the different boundary complexities introduced by the different separations in the five datasets. However, the measured boundary separation is also dependent on the

number of cases available to define the boundary.

The boundary length metric was expected to give a fixed value representing the fixed length of the boundary in all five problems. The values obtained are irregular but appear to give a guide to boundary length that is independent of enforced separation and not overly dependent on the number of cases. The results show this measure to be a rough estimate rather than an accurate measurement of boundary length.

The boundary case density metric correlates well with accuracy in that it rises initially as the number of cases grow and then stabilises at a maximum value. However, as expected, density does not provide a good measure of complexity on its own as it can not discriminate between the different inherent complexities of our five artificial problems.

In developing this approach of identifying and measuring boundary conditions directly, several problems have become apparent, even in the simplified artificial dataset used for testing. While there appears to be some correlation between accuracy and the individual boundary metrics, it is not clear how these metrics should be combined to provide a single measurement of complexity. Likewise, the experiments have made measurements for a single boundary, and it is not obvious how measurements from numerous decision boundaries should be combined to give values for a complete case base.

Scaling up this approach to real problems also introduces several additional difficulties.

- In real data sets, as the problem space gets more complex with increasing dimensionality, a large number of cases in the case base can be identified as boundary cases and a large number of decision boundaries are formed.
- It is difficult to get accurate measurements of boundary separation where there is an overlap between the groups at the decision boundary and it is not possible to determine if the value obtained refers to a margin of separation or is merely the distance to a case's nearest neighbour belonging to a different class within an overlap condition.
- Boundary length provides only a rough estimate. The positioning of cases at the decision boundary has a large influence on the value obtained and consistent results are difficult to obtain. In addition, finding the minimum distance through the boundary cases is akin to the travelling salesman problem and not easy to solve. As the num-

ber and complexity of decision boundaries increases with real data sets calculating this measure in a multi-dimensional problem space will become impractical.

Explicit modelling of the configuration of a boundary appears to give an indication of the complexity of a problem, both as a result of its inherent complexity and the cases available, on these simple artificial problems. However, scaling up this approach to real problems does present serious difficulties. Another limitation is that local information is only provided about the boundary cases, whereas maintenance policies may require information at a case level about all cases in the case base.

4.2 Complexity Approach

Experimental work carried out on the boundary measure approach in the previous section has identified problems in attempting to directly measure and combine the individual components of problem complexity at a decision boundary. We require a more tractable approach that provides an overall measure of problem complexity but one that also provides individual case information.

Our objective is to help the knowledge engineer make decisions on maintenance strategies by providing a global case base measure of accuracy, noise and redundancy plus local information on the structure of the case base. Our approach is to provide a profile of a local case metric. We use a case complexity measure to provide the local measure and a ranked profile of this measure to provide a view of the overall effect within the case base. The complexity profile identifies the mix of local complexities. In the rest of this section we first define the local case complexity measure used and then look at our profiling approach to providing a global picture of the case base.

4.2.1 Case Complexity

We have chosen an approach that allows us to measure the local complexity based on the overall distribution of cases rather than on specific decision boundary metrics. The building block of our approach is a local complexity measure calculated for each case. In this approach the complexity of each case is calculated based on the class composition

of a case's neighbours while incrementally increasing the size of the neighbourhood being considered. The case complexity identifies areas of uncertainty within the problem space.

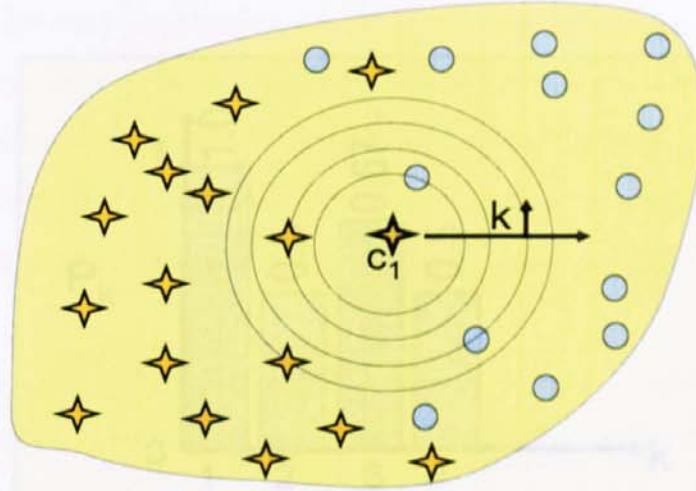


Figure 4.6: Class composition of a cases's neighbourhood

In order to calculate case complexity we first consider the local mix of solutions present in the region of a case, c_i . $P_k(c_i)$ is the proportion of cases within c_i 's k nearest neighbours that belong to a different class from itself. In Figure 4.6 a case is represented by a symbol on the plot with the class of the case distinguished by the shape; star or circle. If we consider case c_1 , then as the value of k increases, the sequence of $P_k(c_1)$ starts 1, 0.5, 0.67, 0.5.

If we look at Figure 4.7 we see a graph plotting the first 4 values of the nearest neighbour profile for c_1 showing $P_k(c_1)$ as k increases. The case complexity measure is based on the area of the graph under the profile and is calculated by

$$\text{complexity}(c_i) = \frac{1}{K} \sum_{k=1}^K P_k(c_i)$$

for some chosen K : where K is the largest neighbourhood considered. With $K=4$ the complexity of case c_1 is $(1+0.5+0.67+0.5)/4 = 0.67$. The complexity measure is weighted to a case's nearest neighbours, hence, using a large value for K has little impact on the value. However, setting too small a neighbourhood can result in overfitting, as the complexity measure becomes overly influenced by a case's immediate neighbours. We have found that for most case bases $K=10$ gives a balanced size of neighbourhood and is used in our

experiments. The maximum neighbourhood size may be reduced for small case bases or where the number of classes is large and results in small class groupings within the case base.

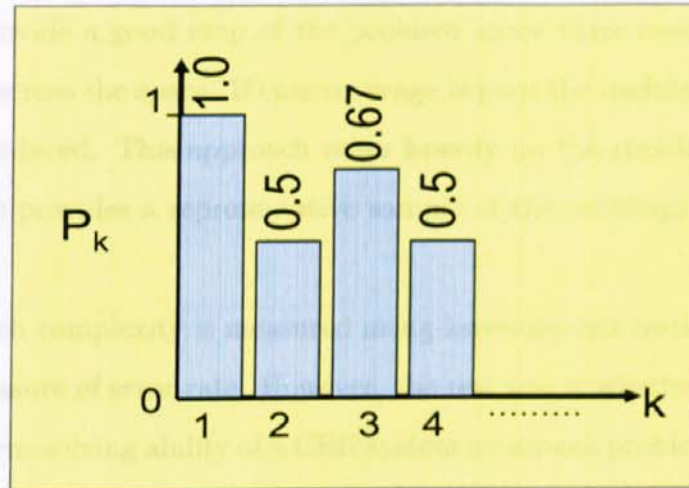


Figure 4.7: Proportion of case c_1 's neighbours belonging to a different class

The complexity of a group or the whole case base can now be calculated by taking an average of the individual case complexities contained in the group or case base.

Discussion of Case Complexity

In this approach, we do not measure the boundary conditions directly but instead use the cases held within the case base to provide a *map* of the local complexities within the problem space. Unlike the boundary approach, the complexity approach provides information on each case that can be used to develop informed maintenance policies. Cases with high complexity are close to classification boundaries and identify areas of uncertainty within the problem space. Cases with complexity greater than 0.5 are generally closer to cases of a different class than those of their own class, and are potentially noisy. Cases with low complexity are surrounded mainly by cases with the same class as themselves, and are located in areas of the problem space in which the system would be more confident in making a decision on the class of a new problem. Cases with a zero complexity value are surrounded by a sizeable group of cases with the same class as itself, and may be considered redundant because other cases in the group would be able to solve new problems in this region of the problem space.

The complexity measure can be used in providing a benchmark for the performance of different case bases within the same domain. In addition, the complexity measure provides a consistent and meaningful measure across different problem domains. As a result, the measure can also be used to provide a benchmark across different problem domains.

In order to provide a good map of the problem space there needs to be a reasonable coverage of cases across the space. If case coverage is poor the usefulness of the complexity measure will be reduced. This approach relies heavily on the standard CBR assumption that the case base provides a representative sample of the problems that the system will face.

In this approach complexity is measured using leave-one-out testing and it is expected to give a good measure of error rate. However, the real test is whether complexity reliably predicts the problem-solving ability of a CBR system on unseen problems. An evaluation of the relationship between complexity and unseen test set accuracy is described in Chapter 6.

4.2.2 Complexity Profiling

The complexity measure provides a local indicator of uncertainty within the problem space and we will show that it is useful for informing maintenance algorithms. However, it is difficult for the knowledge engineer to use this local information directly to gain an insight into the structure of a case base from a global perspective. Our approach to providing the knowledge engineer with meaningful access to this pool of local information is to present the data as a ranked profile of case complexities. In this approach the mix of complexities within the case base can be viewed as a profile allowing comparisons to be made between case bases. In addition to providing the knowledge engineer with a global measure of complexity, the profile also provides a global measure of the level of noise and redundancy within the case base.

Creating a Profiling

The ranked complexity profile is created by first calculating the complexity of each case. The cases are then ranked in ascending order of complexity. Then, starting with cases with the lowest complexity, case complexities are plotted against the proportion of cases used. Thus the x-axis shows a case's normalised position in the ranked list and the y-axis

gives the complexity value for a particular case. A typical profile plot, for a case base containing redundancy, is shown in Figure 4.8. Cases with zero complexity are plotted first on the left hand side of the graph followed by a rising curve as the plot breaks away from the x-axis as the complexity of the cases increases.

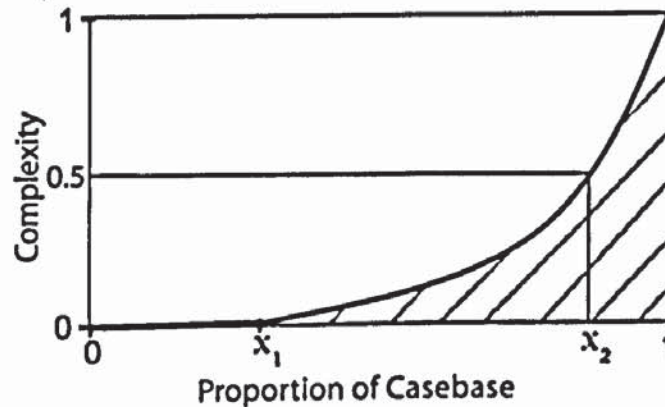


Figure 4.8: Typical graph of local complexity profile

Three key global indicators can be taken from this plot to give a measure of accuracy, redundancy and noise respectively, as follows:-

- **Error Rate:** The area under the curve, shown as the shaded area on Figure 4.8, gives the overall complexity of the problem being faced and provides a measure of expected error rate. This is equivalent to the average case complexity of the cases in the case base.
- **Redundancy:** The position at which the plot breaks away from the x-axis, shown on the profile as x_1 , gives a measure of the level of redundancy within the case base. This is a measure of the proportion of cases located in single class clusters.
- **Noise:** A case with a complexity greater than 0.5 has in most of its neighbourhoods the majority of its neighbours belonging to a different class. These cases can be considered noisy. The proportion of noisy cases can now be portrayed as $1 - x_2$, the distance to the right of x_2 .

The area under the curve provides additional information. It gives a measure of the positioning or contribution of the cases within the case base. If the case base contains

many cases contributing little to the classification task a typical exponential curve, as discussed above, will be seen. However, if all the cases are positioned well and contributing more evenly to the classification task a more straight line graph is expected as shown in Figure 4.9. This would be the expected profile after a redundancy removal editing algorithm has been applied to a case base.

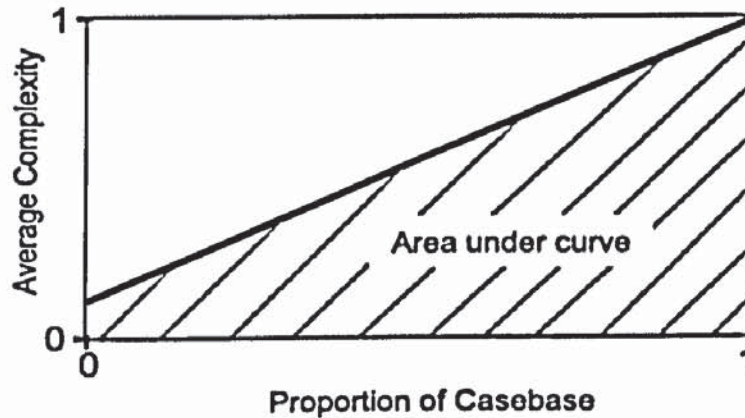


Figure 4.9: Typical complexity profile for an edited case base

The approach described here gives a measure of whether the number of cases in the case base is appropriate to the domain and its level of complexity. It is expected that the three indicators described here will correlate well with typical measures of error rate, redundancy and noise. This will be investigated further in Chapter 6. However, it is the graph itself that provides the best insight into the structure of the case base, and allows informed decisions to be made by the knowledge engineer in relation to whether the number of cases in the case base is appropriate to the domain and its level of complexity.

Comparison of Case Base Profiles

We looked at a *typical* complexity profile and claimed that this profiling provided a good approach at making comparisons across different domains. To examine this claim we look at example complexity profiles from four domains. Figure 4.10(a)-(c) show the complexity profiles for three public domain classification datasets from the UCI ML repository (Blake et al. 1998), together with the complexity profile for an artificial dataset in Figure 4.10(d).

Wine in Figure 4.10(a) is a simple three class problem with 14 numeric attributes and 178 instances. It can be seen from the profile that a high level of classification accuracy is

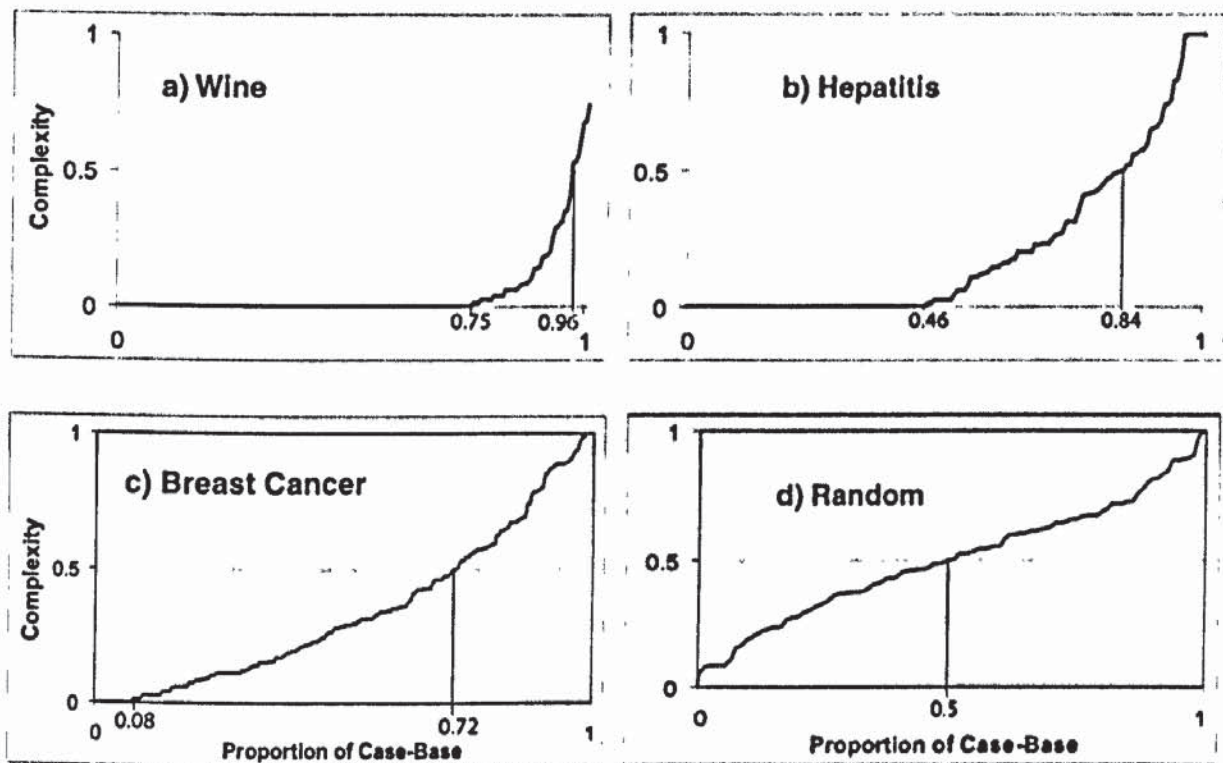


Figure 4.10: Complexity profiles for sample datasets

expected due to the small area under the complexity curve (0.05). The expected level of noise is very low at 4% with the maximum complexity value being well below 1. A high level of redundancy is also evident with 75% of the instances having a zero complexity value. A case base created from this dataset, containing less redundancy, could form part of an excellent CBR problem-solver because the similarity measure forms the instances into clusters with the same solutions - similar problems have similar solutions.

Hepatitis (Figure 4.10(b)) is a smaller dataset of 155 instances represented by 20, mostly nominal, features containing some missing values. This is a more complex problem with an overall complexity of 20% and a gentler slope to the curve than for Wine, suggesting more complex decision boundaries. There is a moderate predicted level of noise (16%) but also several instances completely surrounded by instances of an opposing class with a complexity value of 1. Although there is less redundancy than for wine, the level is still high with 46% of the instances surrounded by at least 10 instances with the same class. Applying noise reduction algorithms would probably improve the level of accuracy achieved and redundancy reduction algorithms could be applied to reduce storage

requirements without affecting accuracy levels.

Breast Cancer in Figure 4.10(c) is a binary classification domain with 9 multi-valued features containing missing data. This is a complex problem, with the low slope on the graph indicating most instances lie close to decision boundaries. There is a high estimated level of noise (28%) and little redundancy (8%). This profile would suggest a dataset that is not suitable for a CBR application as it stands. Applying noise reduction algorithms may improve accuracy levels. In addition, improvements in the similarity measure or case representation could be investigated to create a design in which problems with similar solutions are better recognised as being similar.

The final profile, Figure 4.10(d), is for an artificial dataset with 100 instances. This is a binary classification problem with 2 numerical features where the class of an instance is randomly selected. This is a problem that has been created so that similar problems will not form into cluster of instances with similar solutions. The dataset would not make a suitable case base for a CBR problem-solver and this is confirmed by the complexity profile. As expected, the predicted error rate is 50% and the predicted noise level is also 50% because instances are as likely to be surrounded by instances of an opposing class as the same class. There is no redundancy because the instances do not form into large same class clusters.

4.3 Chapter Summary

In classification tasks the condition at class boundaries largely determine the complexity of problem being faced. The Boundary approach to case base modelling attempted to explicitly measure boundary separation, length and case density by identifying boundary cases related to each decision boundary. The results of initial investigation were promising but gave problems in combining the measures and scaling up the approach to more complex, real-life domains.

A complexity measure has been developed that implicitly measures the level of complexity within local areas of the problem space by looking at the class composition of a case's neighbours. A ranked profile plot of this measure has been used to provide a model of the case base that supplies the knowledge engineer with global information on

accuracy, noise and level of redundancy. The complexity profile also gives an insight into the structure of the case base by making the mix of case complexities within the case base transparent to the knowledge engineer.

Our complexity model is certainly concerned with estimating accuracy and providing local case information. However, our model does more. Complexity profiling can play a further role in assisting the knowledge engineer to make choices between alternative maintenance techniques depending on the structure of the data and a system's performance requirements. Profiling also provides the opportunity to create a benchmark for comparison with future versions of the case base to monitor the impact of changes over time or to evaluate alternative system designs.

The use of our complexity model to maintain the case base by informing specific maintenance algorithms is discussed in the Chapter 5 while the ability of the complexity model to provide accurate predictions about accuracy, level of redundancy and level of noise is evaluated in Chapter 6.

Chapter 5

Complexity-Guided Case base Maintenance

The retention stage of the CBR process is now considered to involve far more than the act of incorporating the latest problem-solving experience into the case knowledge. Case base maintenance is an integral part of the CBR process (Lopez de Mantaras et al. 2006).

The objective of the complexity-guided case base model was two-fold: to provide the knowledge engineer with a global overview of the case base, and to inform specific maintenance algorithms. We have seen how the model can give the knowledge engineer an insight into the structure of the case base and can allow comparisons to be made between alternative case bases or problem domains. The knowledge engineer can use this insight to determine appropriate maintenance approaches for the specific case base being considered. Now we will investigate how the model can be used to inform maintenance algorithms.

Maintenance may be required for many reasons, for example, to improve the system competence, to reduce retrieval times, or to reduce memory storage requirements. Whatever the objectives there are three common types of maintenance that can be applied to a case base to help achieve them:-

- **Case Discovery** - is required where system accuracy is low because the case knowledge is sparse due to a shortage of cases. These characteristics would typically be represented on a case base complexity profile by low levels of redundancy and a large

area under the curve with high levels of complexity over the whole profile.

- **Redundancy Reduction** - is required where retrieval is slow and memory storage requirements are high due to many *similar* cases being present in the case base. The level of redundancy is apparent on a complexity profile by the high proportion of cases with zero or low complexity.
- **Error Reduction** - is typically required where accuracy is harmed by the presence of noisy cases in the case base. The need for error reduction would be visible on the profile by the proportion of cases with high complexity.

We claim our complexity-guided model can inform specific algorithms to accomplish the maintenance task. In this chapter we examine this claim and introduce new algorithms to perform the common case base maintenance tasks. In Section 5.1 two new case discovery algorithms are introduced. We develop a redundancy removal algorithm in Section 5.2 and discuss a new approach to error reduction in Section 5.3.

5.1 Case Discovery

The availability of cases is crucial to a system's performance because the case base is the main source of knowledge in a CBR system. It is often the availability of existing data to form cases that supports the choice of CBR for a particular problem-solving task. Commercial systems generally assume that a suitable case base already exists and give the case author little help in building the initial case base. However, in real environments there are often gaps in the coverage of the case base because it is difficult to obtain a collection of cases to cover all problem-solving situations.

Adaptation knowledge can be used to provide solutions to new problems that occur in the gaps that result from a lack of case coverage. However, gaining effective adaptation knowledge may require considerable knowledge acquisition effort. The inclusion of additional, strategically placed cases can provide a more cost-effective solution. This presents a more complex challenge when compared to the more commonly researched case base editing or selective sampling problems that have a pool of existing cases from which to

select cases (Wiratunga, Craw & Massie 2003). In contrast, the task of case discovery is to add to the case knowledge using implicit information held within the case base.

A typical graph of system classification accuracy, measured as the case base size increases with the addition of cases from the retain stage of the CBR cycle is shown by the solid line in Figure 5.1. It can be seen that classification accuracy initially increases steadily to a maximum value (y_1) with a case base size of x_2 . This is the development stage of a case base in which gaps in coverage are gradually filled by the addition of cases. As the case base size continues to increase accuracy remains relatively stable. This is the mature stage of the case base in which the addition of cases has little effect on accuracy. Traditional case learning can be a slow process and case discovery can assist the case author during the crucial case base development stage by actively identifying *useful* new cases to fill gaps that exist in coverage. Case discovery should result in a shift in the curve to the left, during the development stage of a case base (shown by the dashed line). Little impact would be expected on a system's maximum accuracy as a result of case discovery, however, the number of cases at which the maximum accuracy is reached is fewer (x_1 as opposed to x_2).

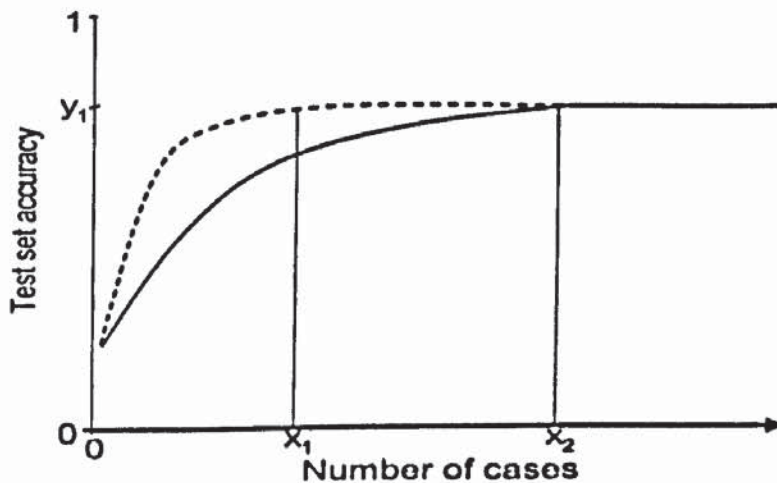


Figure 5.1: Typical graph of test set accuracy as a case base grows

We argue that gaps in the coverage of the case base are in regions of the problem-space in which the system is uncertain of the solution.

5.1.1 Complexity-Guided Approach

Our aim is to discover cases that improve the CBR system's accuracy. We believe cases close to classification boundaries are most likely to achieve this aim. The case discovery problem can be considered in two stages: the identification of *interesting* areas of the problem space in which to place new cases is discussed in this section, while the creation of new cases to fill these gaps is discussed in the following section.

Previous research on case base editing has highlighted the importance of cases in boundary regions for the competence of a case base (Brighton & Mellish 2002, Wilson & Martinez 2000). It seems reasonable to expect a successful case discovery algorithm to also identify cases on class boundaries. Our approach to identifying where new cases should be placed, in order to improve a system's accuracy, involves several stages that combine to identify boundary cases.

Areas of Uncertainty

The first step in finding interesting areas for new cases is to find areas in which cases are more likely to be wrongly classified. We use our case complexity measure, defined in Section 4.2, to identify these areas. This approach gives a measure of the local complexity based on the spatial distribution of cases rather than on a probabilistic distribution. Cases with high complexity, which we refer to as target cases, are close to classification boundaries and identify areas of uncertainty within the problem space. The regions around these target cases are identified as requiring support. Target cases are ranked in descending order of complexity to prioritise between the different regions of the problem space.

Class Boundaries

Target cases identify regions of the problem space, near classification boundaries, that would benefit from the support of additional cases but give no help as to where within these regions the new cases should be placed. Following our hypothesis that cases close to class boundaries are important in case discovery we want to discover cases closer to

the boundaries. Tomek (1976b) uses the distance to a case's nearest unlike neighbour (NUN) to rank cases prior to applying Hart's CNN editing algorithm in order to ensure boundary cases are retained. Likewise, Doyle, Cunningham, Bridge & Rahman (2004) use the concept of a NUN to identify cases nearer to decision boundaries that are then used to provide more convincing explanations to support the proposed solution. We use a similar approach. There must be a classification boundary in the problem space close to the target case, however its direction and location are not known. To find an outer limit for the location of the boundary, the target case's NUN is found i.e. the nearest case that has a different class. The boundary lies between the target case and its NUN. Figure 5.2 shows a target case's NUN being identified on a representation of a small case base using our standard nomenclature.

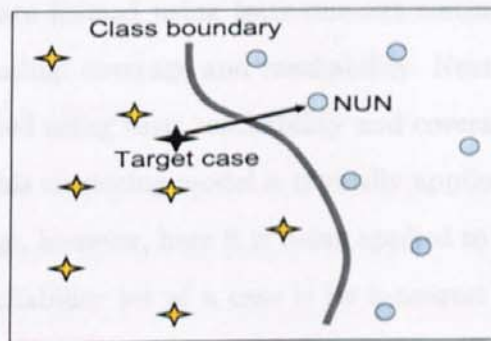


Figure 5.2: A target case's nearest unlike neighbour being identified

Clustering

Prioritising regions of the problem space using only the complexity value of cases is expected to identify interesting areas in which additional cases will help to improve the system's accuracy. However, prioritising on case complexity alone potentially gives two problems. There is a danger that new cases will be concentrated in a small region of the problem space as high complexity cases are likely to be located close to each other. In addition, new cases may be concentrated on small pockets of cases whose classification is different to their neighbours, as these cases will have high complexity values, resulting in poorer performance in noisy or multi-class problems. Figure 5.3 shows five target cases being identified by complexity ranking alone on a representation of a small case base. It can be seen that four of the cases are concentrated in one area of the problem space.



Figure 5.3: Target cases identified without clustering

Partitioning the case base into clusters may give a more balanced distribution of discovered cases over the whole case base. Competence group clustering (Smyth & McKenna 1998) is a commonly used clustering technique in CBR and a similar approach has been adopted here. Clusters are formed using leave-one-out testing to measure the problem-solving ability of a case using: coverage and reachability. Next clusters of cases (i.e. competence groups) are formed using their reachability and coverage sets to group cases that have overlapping sets. This clustering model is typically applied to CBR systems incorporating an adaptation stage, however, here it is being applied to retrieve-only classification. In this scenario, the reachability set of a case is its k -nearest neighbours with the same classification but bound to the first case of a different class (Brighton & Mellish 1999).

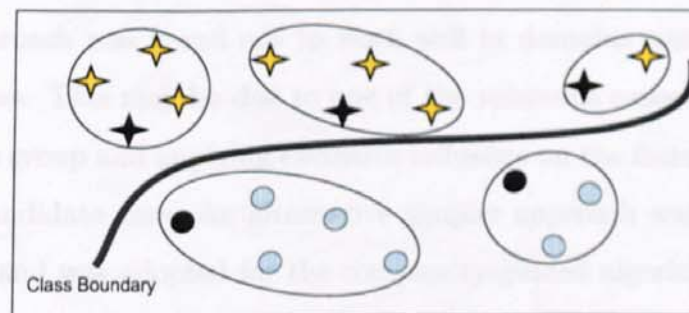


Figure 5.4: Target cases identified with clustering

With the case base formed into clusters, the complexity of each cluster is defined as the average complexity of the cases it contains. The clusters are then ranked in descending order of complexity. Now, rather than choosing target cases purely on complexity ranking, one case can be chosen from each cluster with cluster complexity used to prioritise the target cases. The target case chosen from each cluster is the case with the highest com-

plexity. In addition, there is now the opportunity to remodel the case base, by reforming the clusters as new cases are added, and building the effect of the discovered cases into the next round of case discovery. Figure 5.4 shows the selection of five target cases using this approach. A more even distribution of target cases along the decision boundary can be seen.

5.1.2 Creating a new Case

The approaches described in the previous section are combined to identify gaps in the coverage of the problem space. The second stage of the case-discovery process is to create a *candidate* case to occupy the area between the two reference cases (i.e. the target case and its NUN). This involves setting suitable feature values for the candidate case.

Candidate Case Feature Values

Two approaches for setting the candidate case's feature values have been investigated. In the first, the feature values are set as either the mean (numeric features) or majority (nominal features) of the feature values of the reference cases and their related sets: where a case's related set is the union of its coverage and reachability sets (McKenna & Smyth 2001a). This approach was found not to work well in domains containing small groups of exceptional cases. This may be due to one of the reference cases coming from a much larger competence group and applying excessive influence on the feature values, and hence location, of the candidate case. An alternative simpler approach was found to give more consistent results and was adopted for the complexity-guided algorithms. In this simpler approach the candidate case uses only the two reference cases to set its problem feature values. This results in a discovered case more evenly spaced between the pair of reference cases.

Case discovery aims to create a new case for inclusion in the case base. Inclusion of the candidate case may be automatic but, as there is no guarantee that a candidate case will be a valid case occupying an active area of the problem space, the more likely scenario is for the case author to validate and possibly amend the case prior to its inclusion in the case base.

Noise Filter

A potential problem of discovering cases on classification boundaries is that noisy cases may be discovered in domains containing significant levels of noise. Indeed, most modern case editing algorithms (Brighton & Mellish 2002, Delany & Cunningham 2004) apply noise reduction algorithms prior to an editing approach that retains boundary cases.

A typical approach to noise reduction is to remove cases that are incorrectly classified (Wilson 1972). We apply a similar approach to determine if a validated case should be included in the case base. A *friend to enemy* distance ratio is calculated using the similarity metric. The friend distance is the average distance to the validated case's three nearest like neighbours whereas the enemy distance is the average distance to the validated case's three NUN's.

A high ratio value indicates a validated case that may harm the system's accuracy and would not be included in the case base. A conservative or aggressive approach to noise filtering can be applied by varying the ratio above which a validated case is not added to the case base. When applied, a conservative approach has been taken to noise filtering by not accepting validated cases with a ratio greater than 1.5.

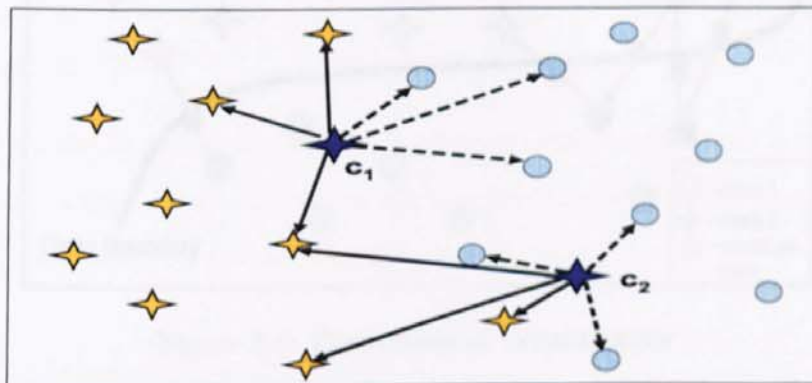


Figure 5.5: *Friend to enemy* distance ratio

Figure 5.5 shows the impact of this friend:enemy ratio on the validation of two candidate cases. The two cases (c_1 and c_2) for which the distance ratio is being calculated are shaded in black. It can be seen that case c_1 has a value for its *friend to enemy* ratio of less than 1, as it is slightly closer to cases of its own class and would not be considered as

noisy. However, case c_2 has a far higher value, in excess of 1.5, because it is close to cases with a different class and would be considered noisy.

5.1.3 Algorithms

Two algorithms have been implemented and tested using the approaches discussed in the previous two sections.

- **COMPLEXITY** is our simpler complexity-guided algorithm. The case complexity measure is calculated for each case and the 50% of cases with the highest complexity are ranked in descending order. Each case in turn (until the desired number of cases are discovered) is selected as the target case and its NUN is identified as its paired case. These two reference cases are used to create a candidate case to lie between them by setting the candidate's feature values as either the mean or majority of the reference cases' feature values. Figure 5.6 shows a simplified representation of the algorithm operating on a small case base with cases belonging to two classes with a class boundary between them.

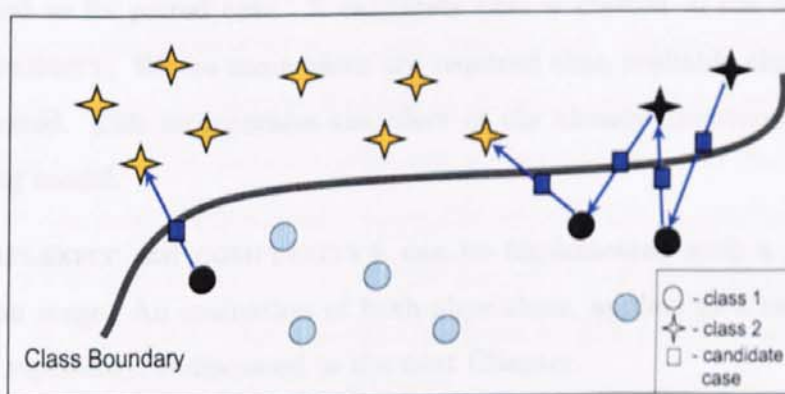


Figure 5.6: Illustration of COMPLEXITY

- **COMPLEXITY+** is a more informed algorithm that uses clustering to create a model of the case base. Figure 5.7 illustrates the operation of the algorithm on a representation of a case base, with cases belonging to two classes and a class boundary between them. The cases are formed into clusters and the case with the highest complexity in each cluster is chosen as the target case (shown as a solid case). The target case's NUN is found (shown by an arrow) giving two reference

cases and a candidate case is created to lie between them, as shown by the square.

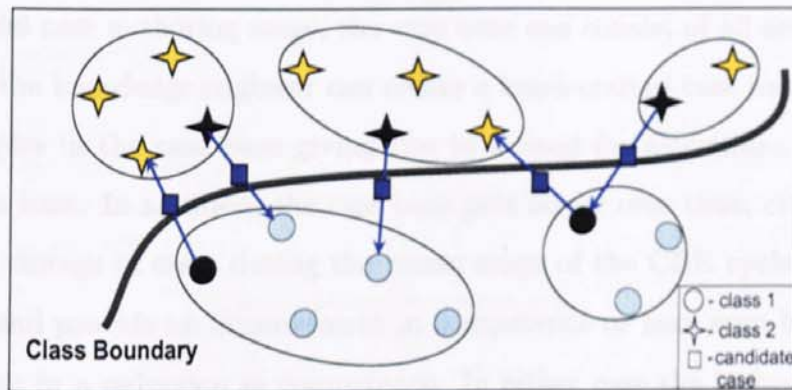


Figure 5.7: Illustration of COMPLEXITY+

The implementation of the algorithm involves the following stages. The case complexity measure is calculated for each case. Clusters are formed and their complexity calculated, as discussed earlier. The 75% of clusters with highest complexity are ranked in descending order of group complexity. A target case is selected from each cluster in turn (until the required number of cases are discovered) and its NUN is selected as its paired case. A candidate case is created in the same manner as in COMPLEXITY. Where more cases are required than available clusters the stages are repeated. This incorporates the effect of the already discovered cases into the clustering model.

Both COMPLEXITY and COMPLEXITY+ can be implemented with a post-processing noise validation stage. An evaluation of both algorithms, applied to a range of datasets from the UCI repository, is discussed in the next Chapter.

5.2 Redundancy Reduction

The CBR paradigm typically employs a lazy learning approach, such as k -nearest neighbour (Cover & Hart 1967), for the retrieval stage of the process which delays generalisation until problem-solving time. This is attractive because training is not necessary, learning is fast and incremental, algorithms are simple and intuitive, and advance knowledge of the problems to be faced is not required (Aha 1997). However, with large case

bases, the drawbacks of lazy learning are high memory requirement since all examples are stored, slow retrieval times and the possible inclusion of harmful cases.

At the initial case authoring stage, the case base can consist of all available examples. Alternatively, the knowledge engineer can create a hand-crafted case base by storing only selected examples in the case base giving rise to a need for algorithms that control the size of the case base. In addition, the case base gets larger over time, often as a result of indiscriminate storage of cases during the retain stage of the CBR cycle. The cases may be redundant and provide no improvement in competence or may even be harmful, noisy cases that result in a reduction in competence. In either case the inclusion of additional cases will increase storage requirements and retrieval times. The cost of retrieval can grow to the extent that it outweighs the benefit of additional cases. This is called the *utility problem* (Francis & Ram 1993, Smyth & Cunningham 1996) and results in an ongoing requirement to control case base growth.

We argue that there is not a single correct answer as to the level of editing a case base requires because a balance has to be struck between the level of compaction and competence, as shown in Figure 5.8. The *best* position on this balance is dependent on a particular system's requirements. In one system fast retrieval times may be vitally important while in another maximum retrieval accuracy may be the most important factor.

Understandably, there has been considerable research on the case base editing problem giving the knowledge engineer a choice of potential approaches. However, most contemporary editing algorithms give no control over the size of the edited case base or the impact on competence. In this section, we use our case base profiling technique to inform a new editing algorithm that gives the knowledge engineer more control of the balance between case base size and competence and also provides a level of explanation of the editing process.

The aim of redundancy reduction is to reduce the case base size while, at the same time, retaining the original competence of the case base. Redundant cases will typically have the following characteristics:-

- They are correctly classified in leave-one-out testing
- They are not required to classify other cases

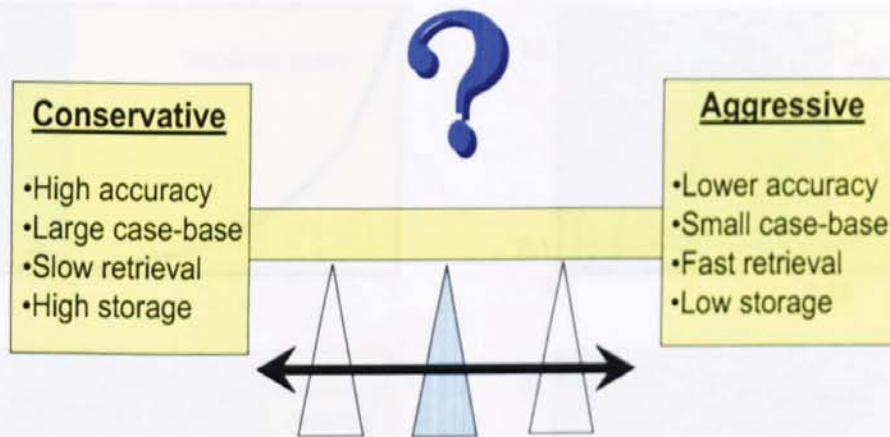


Figure 5.8: Editing algorithms strike a balance between conservative and aggressive editing

- They have larger coverage sets than related sets
- They will be further from class boundaries

Indeed some of these characteristics have been used in numerous editing algorithms to identify redundant cases (Hart 1968, Aha et al. 1991, Wilson & Martinez 1997, Smyth & McKenna 1999a, Brighton & Mellish 2001). Likewise, our approach uses a case complexity measure which encompasses these characteristics to identify redundant cases.

5.2.1 Complexity-Guided Approach

The case base complexity profile provides a tool that can be used for informed redundancy editing in which the knowledge engineer retains control over the level of compaction of the case base. As with most redundancy editing algorithms, our approach aims to give a high classification accuracy and to provide significant storage space reduction. However, these objectives can be contradictory. Aggressive case editing can achieve large reductions in case base size but at the expense of classification accuracy (Delany & Cunningham 2004). The complexity profile provides a measure of the proportion of redundant cases compared to cases near decision boundaries giving an explanation of the effect of different levels of redundancy reduction on competence.

In classification problems redundant cases are found in clusters with the same classification preferably far from decision boundaries. Our approach to case base editing is to identify and delete redundant cases while at the same time retaining boundary cases.

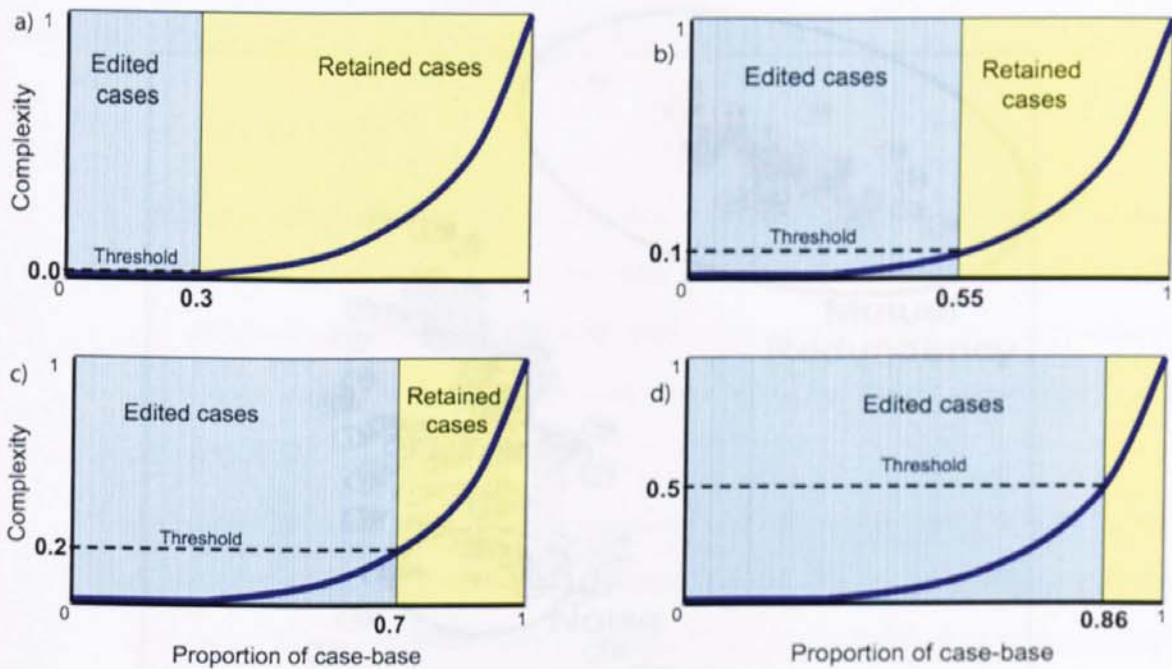


Figure 5.9: Basic threshold approach to redundancy reduction illustrating the impact of different thresholds on edited case base size

The complexity measure, described in Section 4.2.1, is a good identifier of boundary cases, with a high complexity value, and redundant cases with a low complexity. We use the local case complexity to guide our editing algorithm.

The benefits of this approach over existing techniques are two-fold. Firstly, the knowledge engineer is in control of the maintenance process. An informed decision can be made on a suitable level of case base compaction dependent on a system's performance requirements. This decision is not made by selecting an arbitrary case base size. Rather, through a review of the complexity profile, a judgement can be made on the impact of different complexity thresholds. If storage space or retrieval time requirements are crucial to the design a higher threshold can be chosen in the understanding that it will reduce competence. Secondly, the complexity profile provides an explanation of the editing process by providing a transparency to the process and a justification for deleting the selected cases.

The basic approach is to set a complexity threshold and delete cases with a complexity value equal to or less than the threshold. The threshold is set on the y-axis, and the resulting reduction in the size of the case base can be noted on the x-axis. Figure 5.9 gives an illustration of the threshold approach. In Figure 5.9(a) the threshold is set at zero and

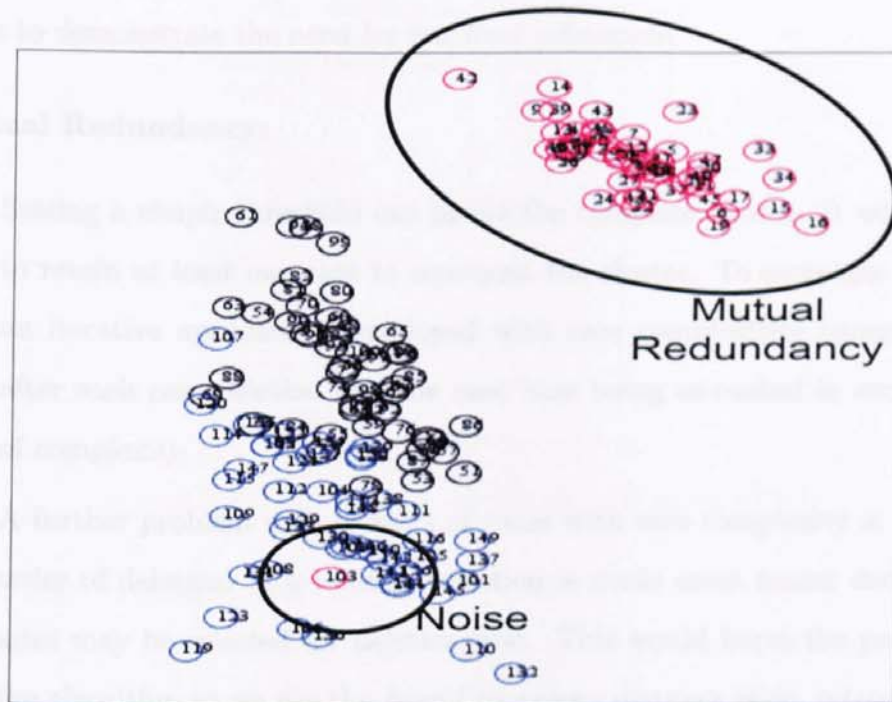


Figure 5.10: Illustration of Iris dataset highlighting need for refinements in relation to mutual redundancy and noise

30% of the cases are edited giving a conservative editing approach. Figure 5.9(b) has a threshold of 0.1 removing 55% of the cases, Figure 5.9(c) has a threshold of 0.2 removing 70% of the cases, while aggressive editing is illustrated in Figure 5.9(d) with a threshold of 0.5 removing 86% of the cases. Our expectation is that setting a zero threshold will remove only cases that are likely to be redundant and not result in a fall in competence. Competence is expected to decline gradually as the complexity threshold is increased.

The basic approach gives promising results but three refinements have been added to improve performance. The thinking behind these refinements is demonstrated by an example. In Figure 5.10 the Iris dataset is displayed on 2 dimensions using a spring model diagram (Kamada & Kawai 1989) with cases being represented by a circle. The diagram attempts to maintain the calculated distance between cases to be proportional to the distance between them on the diagram. Iris is a three class problem and the class is represented by colour: red, blue or black. The red class, highlighted in the figure, forms a distinct cluster of cases all with zero complexity, which we term mutual redundancy. Two refinements have been introduced to counteract the effect of mutual redundancy. Iris is

not a noisy dataset so one noisy case has been introduced artificially by changing the class of one case to demonstrate the need for our final refinement.

- **Mutual Redundancy:**

- Setting a simple threshold can delete the complete cluster. It would be better to retain at least one case to represent the cluster. To overcome this problem an iterative approach is employed with case complexities being recalculated after each case deletion and the case base being re-ranked in ascending order of complexity.
 - A further problem with clusters of cases with zero complexity is the choice of order of deletion. If a random selection is made cases nearer decision boundaries may be selected for deletion first. This would harm the performance of the algorithm so we use the *friend to enemy* distance ratio, introduced in Section 5.1.2, as a secondary ranking. The friend distance is the average distance to the case's nearest like neighbours whereas the enemy distance is the average distance to the case's nearest unlike neighbours. A high ratio indicates a case closer to a decision boundary and farther from cases of the same class. Whereas a low ratio indicates a case farther from a decision boundary and in a cluster of cases all belonging to the same class.
- **Noise:** Noisy cases are by their very nature boundary cases and hence will be retained by this algorithm together with the cases surrounding the noisy case. Adopting the approach of most other contemporary editing algorithms, a pre-processing noise editing algorithm (RENN) is introduced.

5.2.2 Complexity Threshold Editing

The Complexity Threshold Editing algorithm (CTE), incorporating the three refinements introduced above, is described in Figure 5.11. An evaluation of CTE applied to a range of datasets from the UCI repository, is discussed in Chapter 6.


```

T,      Dataset of n cases (c1 ...cn)
COM(S), Calculate case complexity, distance
        ratio and order cases in set S
RENN(S), Apply noise removal to set S
Count=0

COM(T)
For each c in T
  If ( complexity(c)<threshold) count++
End-For
E-Set ← RENN(T)
For 0 to count
  COM(E-Set)
  c ← First case in E-Set
  E-Set ← E-Set - c
End-For
Return (E-Set)

```

Figure 5.11: Complexity threshold editing algorithm

5.3 Error Reduction

In real environments the quality of the cases cannot be guaranteed and some may even be corrupt. Error rates in the order of 5% have been shown to be typical in real data (Maletic & Marcus 2000). Corrupt cases, also called noise, contain errors in the values used to represent the case. In classification tasks, noise can result from either the class labels being wrongly assigned or corruption of the attribute values (Zhu & Wu 2004). The CBR paradigm typically employs k -nearest neighbour for the retrieval stage of the process. While the nearest neighbour algorithm can reduce the impact of noise to some extent by considering more than one neighbour, the existence of noise can still be harmful. This is particularly true where the retrieved cases are being used to support an explanation of the proposed solution (Roth-Berghofer 2004). Manually identifying noisy cases is at best time consuming and impractical with large case bases due to the scale of the task. Hence, automated pre-processing techniques that remove noisy cases are useful to the knowledge engineer during both the initial case base development stage and the ongoing maintenance of a case base.

One of the assumptions underlying the CBR methodology is that similar problems have similar solutions. This assumption is challenged in classification tasks at class boundaries, where the solution changes abruptly as the location of a target case crosses a decision boundary in the problem space. Previous work has identified the importance of boundary

regions for case base maintenance (Brighton & Mellish 1999, Delany & Cunningham 2004). We agree that boundary regions are critical for error reduction. Smoothing the decision boundary by removing selected, *harmful* cases located near boundaries can improve accuracy in some case bases, however, excessive smoothing of the boundary by removing too many cases will reduce accuracy. The *optimal* level of smoothing depends on the characteristics of the decision boundary and is not easily quantified.

Error reduction algorithms aim to improve the competence of a case base by removing cases that are thought to have a detrimental effect on the competence of a CBR system. These cases may be cases that are mislabeled, cases on a boundary between classifications, outlying cases or simply exceptions. Two main approaches have been applied to error reduction:-

1. Remove cases that are incorrectly classified. Wilson editing (Wilson 1972) and several extensions attempt to remove noise by considering each case in the case base and removing it if it is incorrectly classified in leave-one-out testing. Brodley & Friedl (1996) use an ensemble of different type classifiers and use the uncertainty within the results to inform a noise reduction filtering algorithm. Each case is classified using a cross-validation technique and, where there is a consensus among the ensemble a misclassified case is removed. If there is uncertainty in the classification the case is retained.
2. Remove cases that cause other cases to be misclassified. Delany & Cunningham (2004) use leave-one-out testing to identify cases that cause other cases to be misclassified and build what they call the case's liability set (cases where this case contributes to a misclassification). Where a case causes more cases to be misclassified than correctly classified the case is removed i.e. where a cases liability set is bigger than its coverage set. This is a more conservative approach resulting in the removal of fewer cases.

Both these approaches have been shown to successfully remove noise. The criteria used by these algorithms results in the removal of mislabelled cases plus varying levels of boundary cases. However, all these approaches have the disadvantage that they have no control over the level of noise reduction. Each case is simply identified as a noisy case or

not. There is no control over the number of cases removed.

We introduce a pre-processing technique to reduce the error rate in lazy learners by identifying and removing both noisy cases and harmful boundary cases. Our approach identifies potentially harmful cases with the aid of a case base profile and uses a stopping criteria to vary the level of case removal at class boundaries to suit the domain. As an additional benefit, the technique provides an insight into the structure of the case base that can allow the knowledge engineer to make more informed maintenance decisions.

5.3.1 Profiling to Identify Harmful Cases

Our initial objective is to identify potentially noisy or harmful cases. By adopting the basic premise that cases whose neighbours belong to a different class are more likely to be harmful cases, we use a case distance ratio that provides a local measure of a case's position in relation to neighbours of its own class and neighbours with a different class. A ranked profile of this ratio provides a view of the overall structure of the case base. In the rest of this section we first define the local distance ratio used and then look at our profiling approach as a means of presenting a global picture of the composition of local ratios contained within the case base.

Assessing Confidence

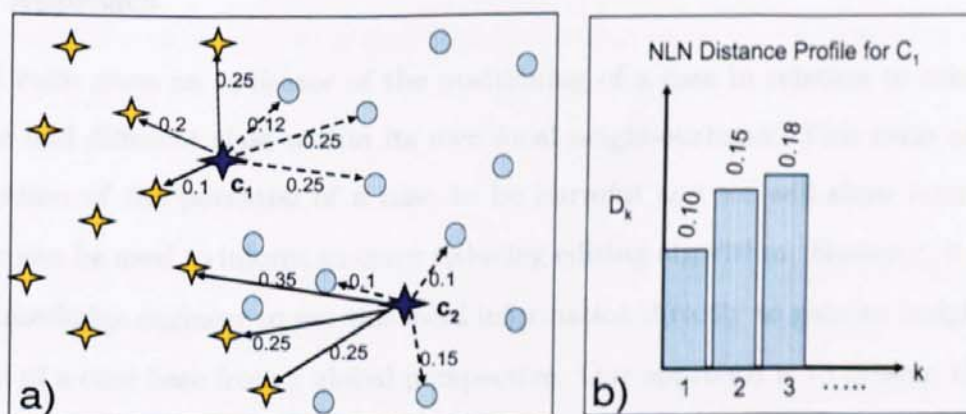


Figure 5.12: Calculation of friend:enemy ratio

The complexity measure we use to assess our confidence in a case compares distances to a case's nearest like neighbours (NLN's) with distances to its nearest unlike neighbours

(NUN's), where the NLN is the nearest neighbour belonging to the same class and the NUN is the nearest neighbour belonging to a different class. We call the complexity measure the Friend:Enemy (F:E) ratio.

Figure 5.12 shows the calculation of the NLN distance ($\text{Dist}(\text{NLN})$) for case c_1 . A case is represented by a symbol with its class distinguished by the shapes: circle and star. Two cases are identified (c_1 and c_2) and the distances to their three NLN's are represented by solid lines and the distances to their NUN's by dashed lines. D_k is the average distance to a case's k NLN's. In Figure 5.12(a), as the value of k increases, the sequence of D_k for c_1 starts 0.1, 0.15, 0.18. A profile of D_k (Figure 5.12(b)) can now be plotted as k increases. $\text{Dist}(\text{NLN})$ is the average value of D_k for some chosen K . For c_1 with $K=3$, $\text{Dist}(\text{NLN})$ is 0.14; $\text{Dist}(\text{NUN})$ is 0.17 and the F:E ratio is 0.83 (0.14/0.17).

C_1 is a typical boundary case, positioned at a similar distance from its NLN's and NUN's, with a F:E ratio in the region of 1. Whereas, C_2 is a typical noisy case, positioned closer to cases belonging to a different class, with a F:E ratio much greater than 1 (2.36). This complexity measure gives a higher weighting to nearer neighbours because they are included repeatedly in D_k and also allows the size of the neighbourhood to be easily varied to suit different sized case bases. A small neighbourhood is typically more suitable for identifying noise and $K=3$ has been used to calculate the F:E ratio for our experiments.

Profile Approach

The F:E ratio gives an indicator of the positioning of a case in relation to other cases of the same and different class within its own local neighbourhood. This ratio can provide an indication of the potential of a case to be harmful and we will show later that this indicator can be used to inform an error reducing editing algorithm. However, it is difficult for the knowledge engineer to use this local information directly to gain an insight into the structure of a case base from a global perspective. Our approach is to present the data as a ranked profile of case distance ratios. In this approach the mix of complexities within the case base can be viewed as a profile allowing comparisons to be made between case bases.

The ranked complexity profile is created by first calculating the F:E ratio of each case to give a local measure of system confidence. The cases are ranked in ascending order.

Then, starting with cases with the lowest value, case distance ratios are plotted against the normalised position of the case in the ranked list. Thus the x-axis shows the proportion of the case base and the y-axis gives the F:E ratio for the particular case at the relative position in the ranked list. A typical case base profile is shown in Figure 5.13.

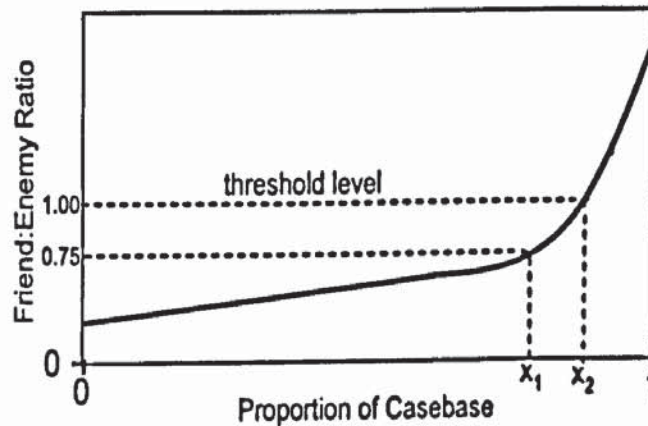


Figure 5.13: Typical graph of Friend:Enemy ratio profile

Two thresholds are marked on the plot corresponding to F:E ratio values of 0.75 and 1.00. The proportion of the case base corresponding to these thresholds is marked as x_1 , and x_2 respectively. These indicators provide an insight into the structure of the case base. The proportion of cases above x_2 identifies cases closer to those belonging to a different class and gives an indication of the level of noise present in the case base. The proportion of cases between x_1 , and x_2 identifies the number of cases close to class boundaries and gives an indication of the potential number of cases that could be removed while smoothing decision boundaries.

Interpreting Profiles

We have looked at a *typical* profile and claimed that these profiles provide a tool for making comparisons of the structure of the case base, including the level of noise, across different domains. To examine this claim we look at example profiles from three domains. Figure 5.14 shows the profiles for three public domain classification datasets from the UCI ML repository (Blake et al. 1998): House Votes, Lymphography and Breast Cancer.

House votes (Figure 5.14(a)) is a binary classification problem with 435 cases represented by 16 boolean valued attributes containing some missing values. It can be seen

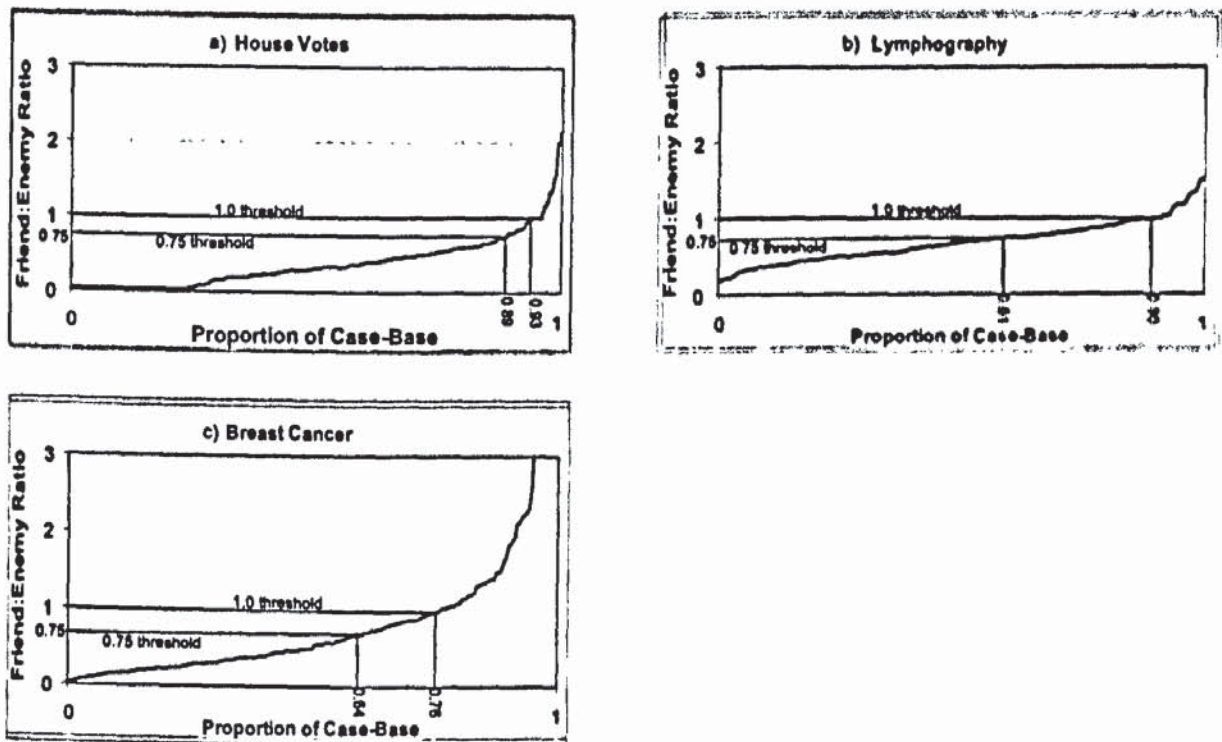


Figure 5.14: Sample profiles for three classic dataset

from the profile that a high level of classification accuracy is expected. There is a low level of predicted noise (7%) based on the 1.0 threshold, and few cases (4%) lie between the 0.75 and 1.0 thresholds, indicating that few cases lie close to decision boundaries. Error reduction techniques would not be expected to give a large improvement in accuracy levels on this case base.

Lymphography (Figure 5.14(b)) is a smaller dataset with 4 classes and 148 cases represented by 19, mostly nominal, attributes with no missing values. There is a low level of predicted noise (10%), however, this appears to be quite a complex problem, with the shape the profile indicating many cases lie close to decision boundaries; 20% of the cases lie between the two thresholds. In this case base it is unclear if smoothing the decision boundary will improve accuracy.

Breast Cancer (Figure 5.14(c)) has 286 cases and is a binary classification domain with 9 multi-valued features containing missing data. There is a high estimated level of noise with 24% of cases with a ratio greater than 1 and a peak F:E value greater than 6. 12% of cases lie between the 0.75 and 1.0 thresholds. Pre-processing to remove harmful cases

would be expected to greatly improve accuracy on this case base.

5.3.2 Complexity-Guided Error Reduction

Our aim in creating an error reduction algorithm is to identify and delete both noisy cases and harmful boundary cases from the case base. Noisy cases are expected to have a F:E ratio greater than 1 while boundary cases are expected to have a ratio in the region of 1. The basic approach we adopt is to set a threshold for the F:E ratio and delete all cases with values above the threshold. An obvious threshold is 1 such that cases positioned nearer to those belonging to a different class will be removed from the case base. However, conservative editing with only limited smoothing of the decision boundaries is possible by setting a threshold above 1 while, conversely, aggressive editing with strong smoothing of the decision boundaries is possible by setting a threshold below 1. In order to establish a suitable threshold across all domains we investigated the effect of setting different threshold values.

Setting the Threshold Level

A ten-times 10-fold cross-validation experimental set-up has been used, giving 100 case base/test set combinations. For each combination, cases with a F:E ratio above the specified threshold were deleted from the case base to form an edited case base. Test set accuracies were recorded on both the original and the edited case bases. The threshold was set at one of fourteen levels between 0.2 and 5. Figure 5.15 plots average test set accuracy on the original case base and on the edited case bases formed with the different thresholds for the three dataset discussed earlier: House Votes, Lymphography and Breast Cancer. Similar patterns of results were observed across other domains.

House Votes shows a small improvement in accuracy as the threshold falls toward 1 but the performance suffers as useful boundary cases are removed with lower thresholds. Lymphography shows no improvement in accuracy from case editing and any boundary smoothing appears harmful. In contrast with Breast Cancer, the accuracy continues to rise until the threshold falls to 0.4, highlighting a domain in which aggressive smoothing of the decision boundaries helps performance.

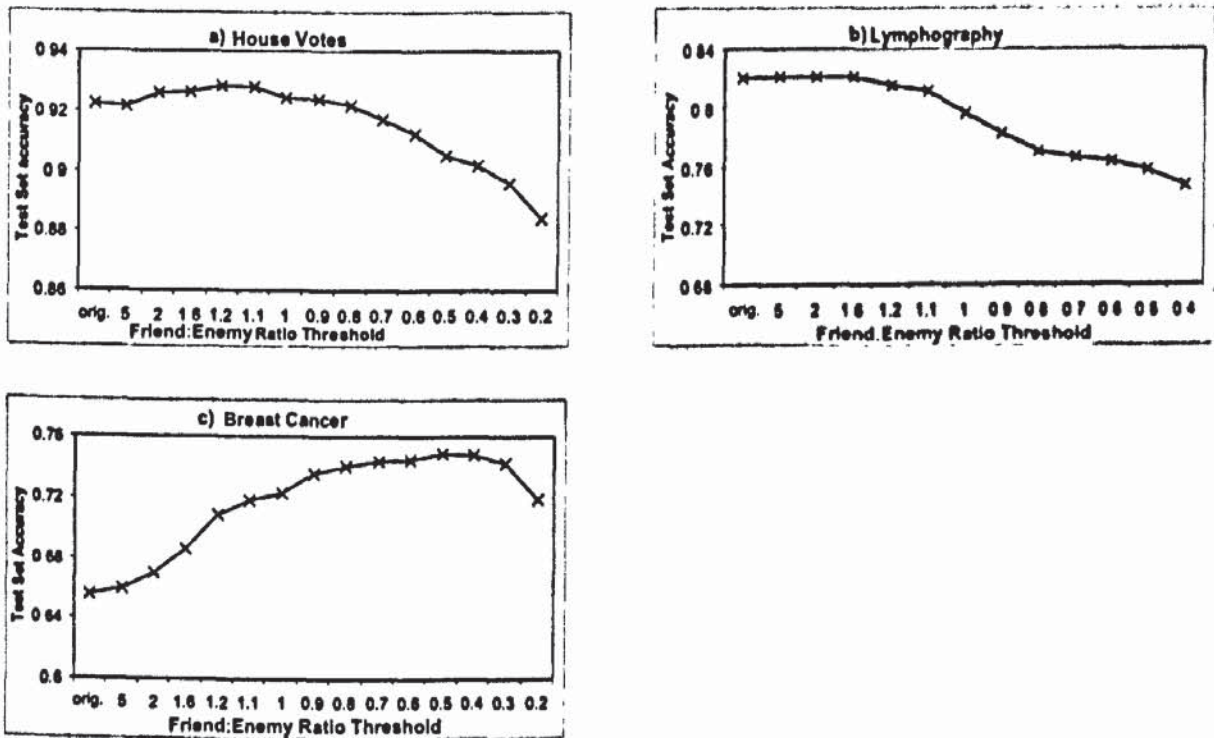


Figure 5.15: Accuracy of edited case bases as cases with ratio above threshold are removed

The expected pattern of results was for the highest accuracy to be achieved with a threshold of about 1. However, this pattern was not observed consistently, as in some domains any boundary smoothing proved to be harmful and reduce accuracy while in others aggressive boundary smoothing with ratio thresholds as low as 0.4 gave the highest accuracy. It is clear that there is not one *optimal* threshold to suit all domains.

5.3.3 Threshold Error Reduction Algorithm

The basic approach, of setting a single threshold, gave promising results in some domains but also highlighted two problems.

- It is difficult to set a single threshold that works well across all domains. It would be better to set a threshold that suits the characteristics of the case base being considered. To overcome this problem and establish an appropriate threshold, we process cases in batches by iteratively reducing the threshold in steps. After each editing step a leave-one-out accuracy check is performed to provide a possible stopping criteria. Leave-one-out accuracy is calculated initially on the original case base and

then on the edited case base after each batch of cases are processed. If the accuracy falls the iterative process is stopped at the present threshold and the edited case base from the previous iteration is accepted as the final edited case base.

- The use of the F:E ratio to identify potentially harmful cases can result in the neighbours of a noisy case being falsely considered to be noisy themselves, simply by the presence of the noisy case in their neighbourhood. This is particularly likely when looking at a very small neighbourhood and weighting the measure to the nearest neighbours as we do with the F:E ratio. To prevent useful cases being mistakenly removed we only delete cases if their complexity is higher than their neighbours. Of course, if a case, not deleted by this check, is truly noisy it will be identified on the next iteration and considered for deletion again.

Our Threshold Error Reduction (TER) algorithm, incorporating the stop criteria and neighbourhood check, is outlined in Figure 5.16. An evaluation of TER applied to a range of datasets from the UCI repository, is discussed in Chapter 6.

5.4 Maintenance Algorithms in Practice

Case base maintenance can take one of two general approaches. The algorithm can operate independently of the knowledge engineer, requiring no input in setting parameters or the scope of the maintenance. For example, many editing algorithms receive a case base as input and output the edited case base without knowledge engineer intervention having the advantage that knowledge of the editing process is not required. The disadvantage is that many maintenance tasks are a trade off between alternative factors that can only be decided with knowledge of the system requirements. In addition, maintenance algorithms are unlikely to be accepted, in a commercial environment, if the process is not transparent.

An alternative approach is to leave control over the scope of the maintenance with the knowledge engineer. This has the advantage that maintenance can be tailored to meet the objectives of the system being developed. The disadvantage is that the approach must be understandable requiring a transparent method in which the impact of decisions can be

```

T-set,      case-base of n cases (c1 ...cn)
COM(S),     calculate F:E ratio, F:E(c), for
             each case in set S
ACC(S),     returns leave-one-out accuracy
             for set of cases, S
CHK(c),     returns true if F:E(c) is > F:E ratio of
             each of its k-nearest neighbours

E-set = T-set
R-set = T-set
accuracy = ACC(T-Set)
threshold = 1.25

while (ACC(E-Set) >= accuracy)
  COM(E-Set)
  for (each c in E-set)
    If (F:E(c) > threshold && CHK(c))
      E-Set = E-Set - c
    endif
  endfor
  If (ACC(E-set) >= accuracy)
    accuracy = ACC(E-set)
    threshold = threshold - 0.1
    R-set = E-set
  endif
endwhile
return R-set

```

Figure 5.16: Threshold error reduction algorithm, TER

judged. We favour the second approach and have developed a prototype interactive tool, called ComCASE, that demonstrates the complexity-guided approaches developed in this research ¹.

The complexity model and maintenance algorithms discussed in this thesis have all been implemented and evaluated on CASE, which is a CBR retrieve only system. CASE has been developed in Java, using an object oriented design to allow easy extension, and provides a work bench on which to test new or existing algorithms. Maintenance algorithms can easily be incorporated with the addition of a new class module. CASE takes an input in .arff file format (Witten & Frank 2000) and has data structures to hold a case base as a collection of cases each composed of a set of features. Features have a name and value and can be allocated one of three types: nominal, ordinal or real. Retrieval is performed by an implementation of the *k*-NN algorithm with a solution by a weighted majority vote for nominal classes, and by a weighted average for real valued classes. The

¹I acknowledge Hassan Khajeh-Hosseini for his work in developing the interface as part of a student summer project which I supervised.

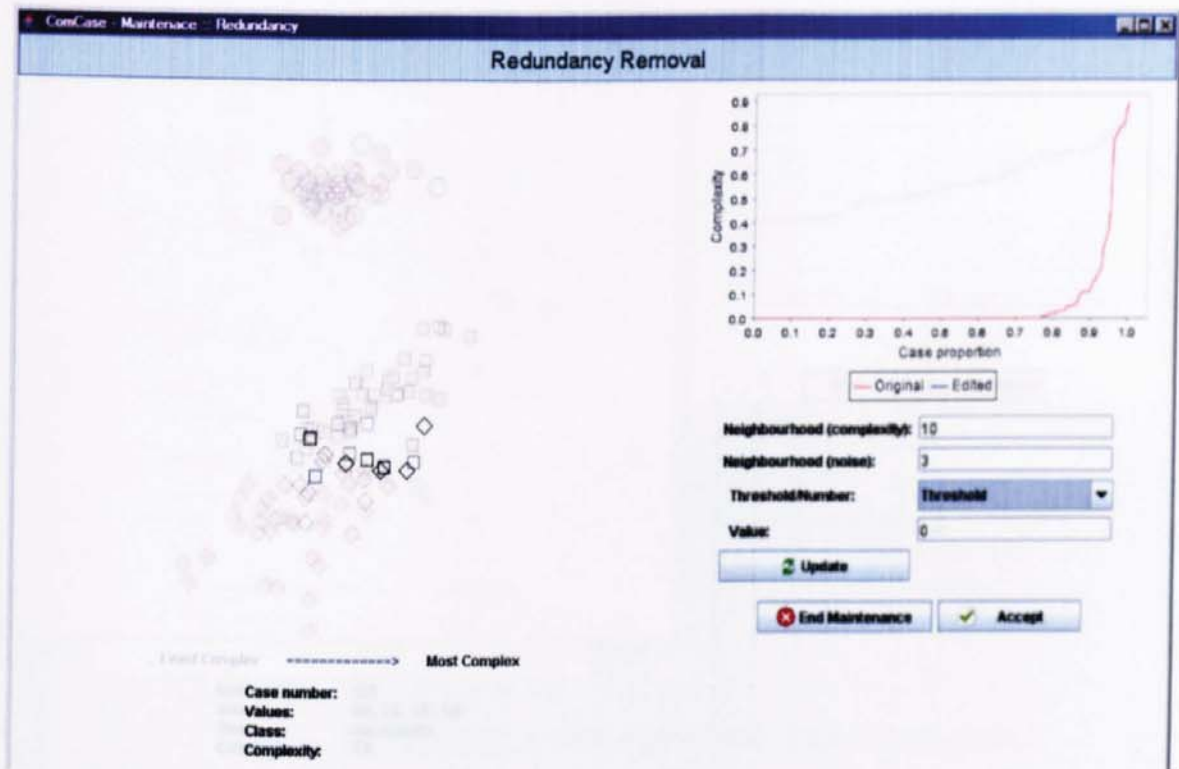


Figure 5.17: Prototype interface for complexity-guided maintenance prior to parameter setting

distance between cases is measured by Euclidean or Manhattan distance, calculated using the normalised distance between feature values. Feature weights can be applied manually or using a genetic algorithm approach if required. ComCASE has been developed as an extension to CASE.

Our general approach is to present the complexity or F:E ratio profile and leave control over the maintenance process with the knowledge engineer. Figure 5.17 shows the initial interface for redundancy reduction maintenance with the CTE algorithm containing three main panels.

- The panel on the upper left of the interface displays a dynamic visualisation of the case base by using a spring based algorithm. The algorithm uses the attraction and repulsion of the *springs* to spread the cases around a two dimensional graph in an attempt to preserve the n-dimensional distances between cases. The class of a case is represented by shape, and the complexity by a colour gradient; light grey for low complexity cases through to black for the most complex cases.

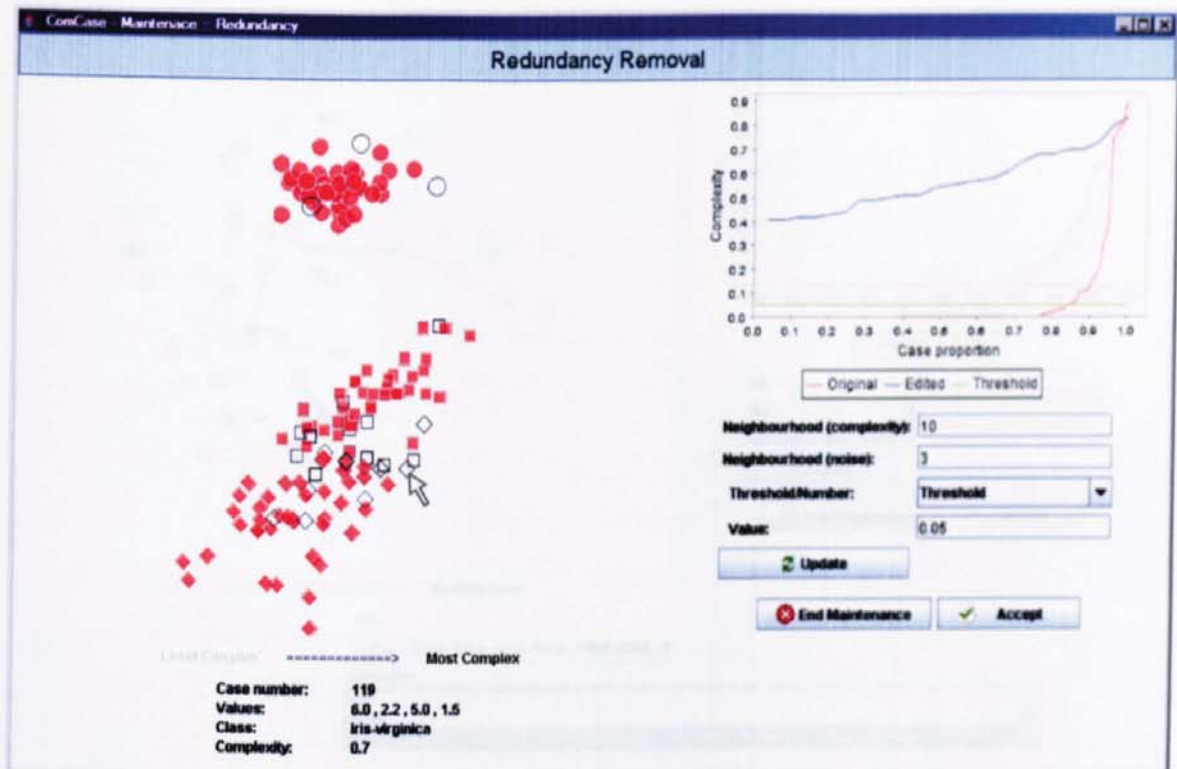


Figure 5.18: Prototype interface for complexity-guided redundancy reduction showing the selected threshold and cases proposed for deletion

In the diagram we see the Iris dataset with the three classes represented by circle, square and diamond. The circle class is well separated in an independent cluster but the square and diamond classes show some overlap at a decision boundary. It can be seen that the darker, and hence most complex cases are located close to the decision boundary as expected.

- The panel at the top right shows the case complexity profile as discussed in Section 4.2. It can be seen from the profile that the Iris dataset has a low overall level of complexity and a high level of redundancy which suggests a redundancy reduction algorithm may be appropriate. While the level of noise is low there are a few cases with high complexity and a slight improvement in accuracy may result from applying an error reduction algorithm.
- The lower right panel contains parameter setting boxes and control buttons.

In Figure 5.18 we see the interface after maintenance parameters have been selected.

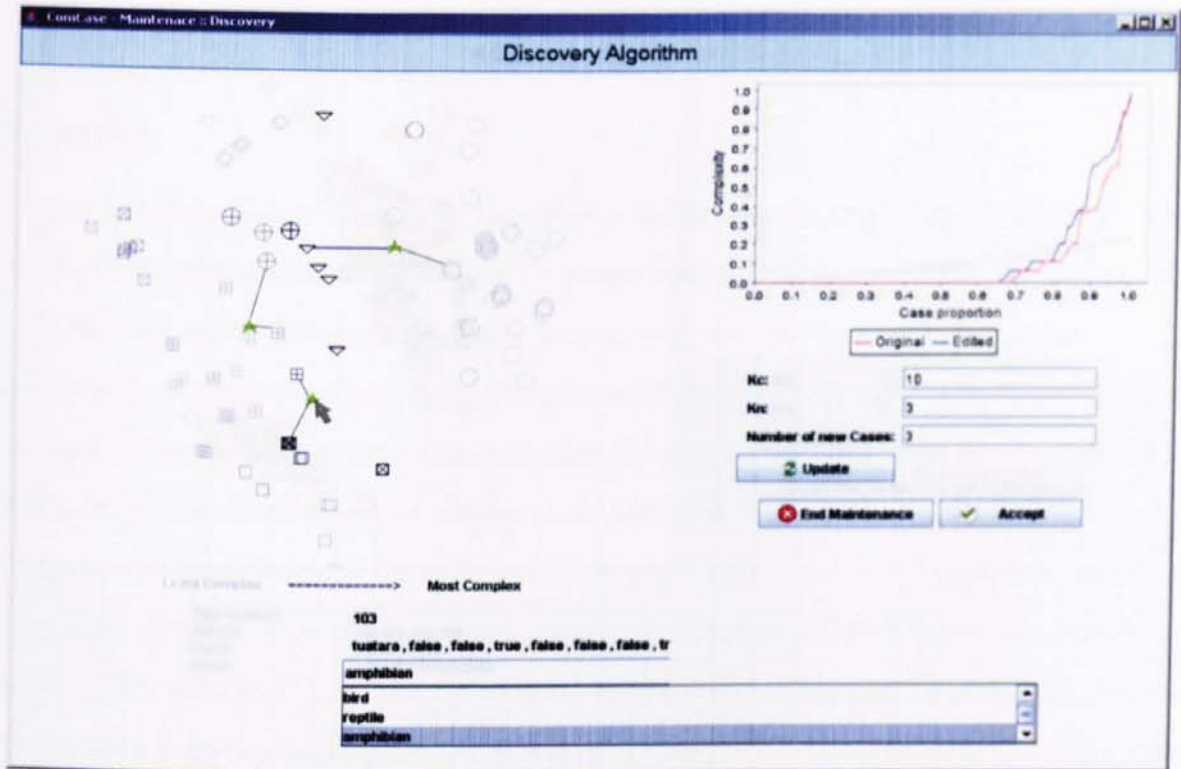


Figure 5.19: Prototype interface for complexity-guided case discovery

In this example a complexity threshold of 0.05 has been selected as can be seen by the green line on the complexity profile. The cases selected by the algorithm for removal have been colour filled in red. A large number of cases are selected reflecting the high level of redundancy present in the case base. It can be seen that the cases retained are mostly close to the decision boundary between the square and diamond class although a few cases are retained to represent the circle class. The complexity profile that results from accepting this editing *proposal* is shown on the profile plot as the blue line. The knowledge engineer has the option to accept this editing proposal in its entirety, to try again with different parameters, or to select and make decisions about individual cases. Case information on selected cases is shown in a panel on the lower left of the profile.

The interface for case discovery is shown in Figure 5.19. The layout of the interface is very similar to that for redundancy removal but with the number of new cases required being the main parameter to set rather a threshold. The example shown is for the COMPLEXITY algorithm being applied to the Zoo dataset. Three new cases have been selected in the parameter panel and can be identified as green stars on the spring diagram. The

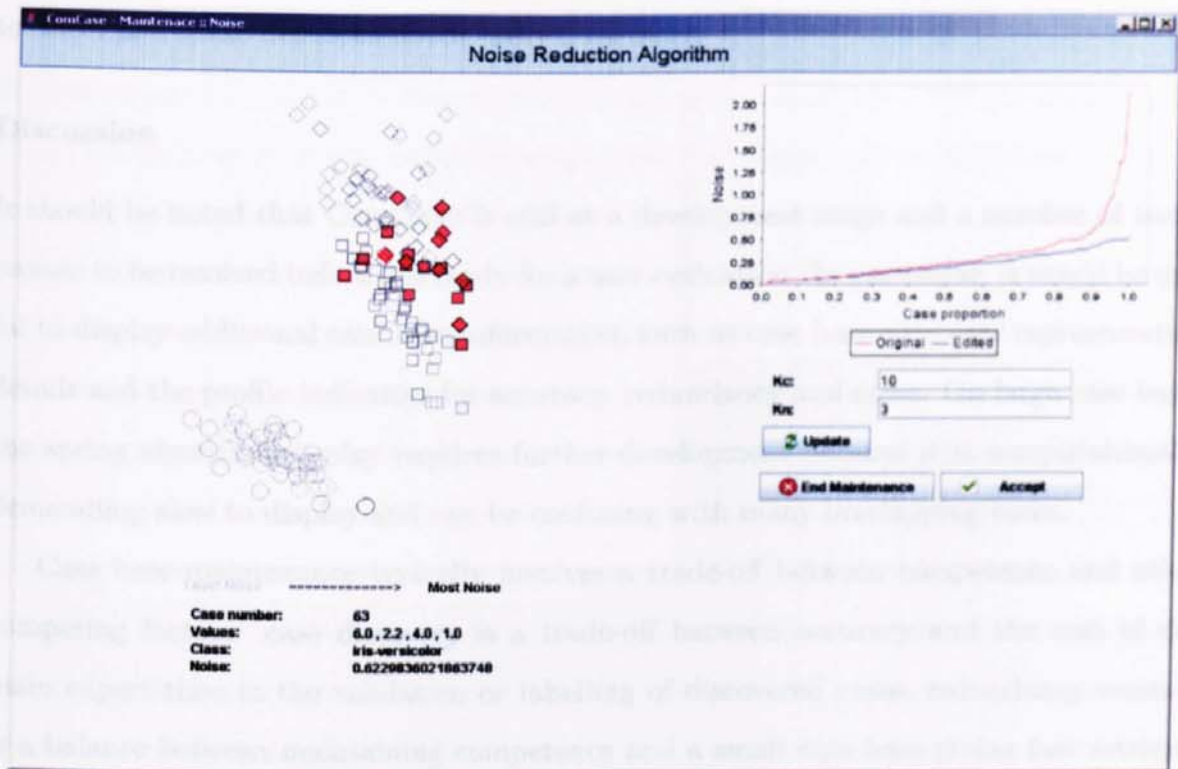


Figure 5.20: Prototype interface for complexity-guided error reduction

reference cases related to the discovered case are identified by a connecting line. While the algorithm does select a class for new cases, this is intended as an interactive approach and the knowledge engineer can identify the new case on the spring diagram and select the appropriate class in the case information panel on the lower left of the interface.

Figure 5.20 shows the interface for applying the error reduction algorithm, TER. In this interface the F:E ratio ranked profile, which uses a smaller neighbourhood and is more discriminating for identifying noise, is shown in the upper right panel rather than the complexity profile shown in the previous interfaces. In the example shown we again see the Iris dataset, however, the individual case positions on the spring based visualisation of the case base have change when compared to Figures 5.17 & 5.18. This highlights one of the disadvantages of the visualisation; it is not deterministic and different diagrams are likely to be produced each time which can be confusing to the user. As expected from the low level of noise present in the data, only a few cases, lying close to the decision boundary, have been selected for removal (shown colour filled in red). The impact of the editing proposal is shown on the F:E ratio profile as the blue line. Again, the knowledge engineer

has the option to accept this editing proposal, to try again with different parameters, or to select and make decisions about individual cases.

Discussion

It should be noted that ComCASE is still at a development stage and a number of issues remain to be resolved before it is ready for a user evaluation. In particular, it would be useful to display additional case base information, such as case base size, case representation details and the profile indicators for accuracy, redundancy and noise. On large case bases the spring algorithm display requires further development because it is computationally demanding, slow to display and can be confusing with many overlapping cases.

Case base maintenance typically involves a trade-off between competence and other competing factors: case discovery is a trade-off between accuracy and the cost of domain expert time in the validation or labelling of discovered cases; redundancy removal is a balance between maintaining competence and a small case base giving fast retrieval times; error reduction is a trade-off between competence and the loss of case knowledge through the deletion of exceptional cases. In developing an interactive approach to case maintenance we aim to leave control of these trade-offs with the knowledge engineer so that informed decisions can be made that consider system requirements. The knowledge engineer can exercise control over the maintenance process by:-

- choosing appropriate techniques based on the information provided on the structure of the case base by the complexity profile.
- setting suitable parameters for particular algorithm e.g. setting an editing threshold for CTE, our redundancy removal algorithm
- accepting, rejecting or adapting the proposed case base as a result of a judgement on its impact, shown on the spring diagram visualisation and on the complexity profile.

One of the main advantages in CBR lies in the transparency of the approach which gives particular advantages when providing explanations to the user (Leake 1996a). Explanation has been a hot topic in CBR in recent years and the focus of considerable research effort (Massie et al. 2004b, McSherry 2004, Nugent, Cunningham & Doyle 2005). However

the research has been aimed almost exclusively at making the reasoning process, its result or the usage of the result understandable to the user (Sormo, Cassens & Annotdt 2005). Similar issues, in relation to the need for explanation, apply to CBR knowledge maintenance but have generally not been considered. If maintenance algorithms are to be accepted by knowledge engineers the process, result and benefit must be explained.

In developing our maintenance approaches we have been conscience that user confidence may be harmed, due to a lack of understanding, if the processes are hidden. We aim to make the maintenance process more transparent by presenting the mix of complexities within the case base as a profile. This gives the knowledge engineer an understanding of the structure of the case base and a measure of the difficulty of the problem being faced. This aids the choice of techniques to apply and gives an indication of the expected result. Using the spring diagram to visualise the case base prior to applying the maintenance technique and to highlight new cases or cases selected for deletion after applying the maintenance technique gives an explanation of the process. The impact of the maintenance process can be judged by the change to the profile and the spring diagram. Explanation is further enhanced by adopting a consistent approach across all three maintenance tasks addressed in this research. The outcome is a transparent approach in which visualisation is used to provide a knowledge-light explanation of the maintenance process and result, thus aiding understanding by the knowledge engineer.

5.5 Chapter Summary

In this Chapter we have looked at how the local case information generated from our complexity-guided model for classification tasks can be used to inform case base maintenance approaches. Specific algorithms have been developed for three common maintenance tasks: case discovery, redundancy reduction and error reduction. A prototype system has been developed which demonstrates the use of the algorithms in practice.

A complexity metric and a case's NUN have been used to guide the case discovery process by identifying interesting areas of the problem space. The idea of placing new cases on classification boundaries appears to be intuitively sensible in that it mirrors the

approach of recently developed case base editing algorithms. **COMPLEXITY** and **COMPLEXITY+**, two new complexity-guided algorithms that adopt this approach, have been introduced.

To identify redundant cases that contribute little to classification performance we combine a local case complexity together with a case base profile to guide the editing process. The complexity measure identifies redundant cases for deletion and cases on class decision boundaries for retention. By leaving control of the threshold parameter with the knowledge engineer, an element of control is retained over the compromise required between the contradictory objectives of the reduction in case base size and the retention of competence.

For error reduction we use an alternative local case distance ratio that considers the distance to neighbours belonging to the same and different classes to aid identifying harmful cases. This measure together with a case base profile guides the editing process for lazy learners. The algorithm (**TER**) focuses on deleting harmful cases from boundary regions to give smoother decision boundaries between classes. A stopping criteria is used to ensure that the level of smoothing is adjusted to suit the domain. Noise reduction can also harm performance. Careful consideration should be given to the domain and the structure of the case base to ensure there is a need for noise reduction before removing case knowledge with an editing algorithm. The F:E ratio profile provides a tool for the knowledge engineer to make an informed decision on the need for error reduction.

ComCASE is an interactive case base maintenance tool that has been developed as an extension of **CASE** to demonstrate our complexity-guided approach to maintenance. **ComCASE** implements the complexity model together with our new maintenance algorithms: **COMPLEXITY**, **COMPLEXITY+**, **CTE**, and **TER**. The interface developed for **ComCASE** displays the complexity profile and a two-dimensional visualisation for both the original and the proposed case bases but leaves control of acceptance of proposed changes with the knowledge engineer.

Chapter 6

Evaluation

We carry out an experimental evaluation of the complexity-guided case base model and associated maintenance algorithms introduced in the preceding two chapters. All the experiments are carried out by implementing and adding the appropriate modules to CASE, the experimental CBR retrieve only test bed described in section 5.4. The objective of the evaluation is two-fold. First to show that the model provides an accurate global view of the case base thus aiding the knowledge engineer to make informed maintenance decisions and giving confidence in the local case information used to inform our new maintenance algorithms. Second to establish the performance of the new maintenance algorithms in relation to existing benchmark algorithms.

In the following section the data used in the experimental evaluation is described. In Section 6.2 the complexity profile indicators are examined to determine their ability to predict the real dataset values. The new case discovery algorithms are experimentally evaluated in section 6.3 to establish whether *useful* new cases are discovered. In sections 6.4 and 6.5 the new redundancy reduction and error reduction editing algorithms are investigated and their performance compared with existing benchmark algorithms.

6.1 Datasets

Seven public domain classification datasets from the UCI ML repository (Blake et al. 1998) have been used in the evaluations reported in this Chapter. The selected datasets have been chosen to provide varying number of cases, features and classes and differing

proportions of nominal to numeric attributes. Table 6.1 gives a summary of the datasets used including a measure of the difficulty of the classification problem in the form of test set accuracies achieved with three standard classifiers¹. Some of the datasets are recognised to be noisy, e.g. Breast Cancer, and present more difficult problems while others, e.g. Wine, have no or low levels of erroneous data and present relatively easy problems with accuracies of over 95% . Some of the datasets contain missing data and column 6 shows the number of attributes which contain missing data.

Table 6.1: Comparison of UCI datasets used for evaluation

CASE BASE	No. of CASES	No. of CLASS	NO. OF ATTRIBUTES			CLASSIFIER ACCUR. %		
			NOMINAL	NUMERIC	MISSING	1-NN	J48	N.BAYES
Breast Cancer	286	2	9	0	2	72.4	75.5	71.7
Hepatitis	155	2	13	6	15	80.7	83.9	84.5
House Votes	435	2	16	0	16	92.4	96.3	90.11
Iris	150	3	0	4	0	95.3	96.0	96.0
Lymphography	148	4	15	3	0	82.4	77.03	83.11
Wine	178	3	0	13	0	94.9	93.8	96.6
Zoo	101	7	16	1	0	96.0	92.1	95.1

6.2 Complexity Model

Our complexity model profiles can be evaluated on two levels: whether complexity profiling can provide useful comparisons of case bases from different domains; and, secondly, whether the profile indicators accurately predict global error rates and levels of noise and redundancy.

In Section 4.2.2 we looked at profiles from different domains, discussed the insight these profiles provide and demonstrated how they could clearly identify the differences between example datasets. However, the assessment of the profiles assumes that the error rate, noise level and redundancy level indicators are good predictors of the real values contained within the data. While conceptually the use of these indicators appears reasonable, we want to investigate the relationships empirically. In this section of our evaluation we aim

¹Classification accuracy is measured using standard 10-fold cross validation parameters with the WEKA machine learning workbench (Witten & Frank 2000)

to confirm our assertion that the complexity profile indicators accurately predict global error rates and levels of noise and redundancy.

Table 6.2: Results summary of complexity profile indicators compared to alternative measures

Case Base	ERROR RATE		NOISE		REDUNDANCY
	TEST SET	PROFILE	ENN	PROFILE	PROFILE
Wine	0.037	0.050	0.033	0.04	0.75
Iris	0.059	0.058	0.048	0.05	0.79
Hepatitis	0.189	0.203	0.176	0.16	0.46
Lymphography	0.187	0.242	0.155	0.14	0.23
Breast Cancer	0.339	0.344	0.306	0.28	0.08
House Votes	0.079	0.083	0.071	0.07	0.77
Zoo	0.038	0.085	0.061	0.06	0.70

Accuracy or error rate is the easiest indicator to compare. We measure error rate experimentally using ten-fold cross-validation. Nine folds are retained as the training set with the remaining fold being the unseen test set. The average error rates for seven UCI datasets, calculated using 1-NN, are shown in column 2 of Table 6.2 with the corresponding error rate indicator from the complexity profiles shown in column 3. In Figure 6.1 the average test set error rate, measured for the seven datasets, is recorded along the x-axis with the corresponding error rate indicator from the profile is recorded on the y-axis. There is a strong correlation between the results as can be seen by the close fit to the straight line.

There is not an obvious measure of noise with which to make a comparison. However, ENN is one of the best known noise reduction algorithm, hence, we use the reduction in the size of a dataset after applying ENN as a benchmark measure of noise with which to compare our predicted indicator from the complexity profile. The average reduction in the edited set size, after applying ENN, as a proportion of the original dataset size is shown in column 4 of Table 6.2. This is compared with the average complexity profile noise indicator, shown in column 5. Again there is a strong correlation between the results, shown by the fit to a straight line in Figure 6.2 which plots the complexity profile noise prediction on the y-axis with the proportional reduction in the size of the dataset from applying ENN on the x-axis.

These results confirm that the complexity profile indicators are good predictors of accuracy and noise. The ability of the profile to predict redundancy is more difficult to measure directly but is investigated in more detail in Section 6.4.

To some extent the close correlation between the values predicted by the profile and the real dataset values is to be expected because both the complexity measure and k -NN look at the mixture of solutions present in a cases neighbourhood. However, the actual measurement used and the size of the neighbourhood are certainly different which makes the closeness of the correlation encouraging with respect to the models ability to inform maintenance algorithms.

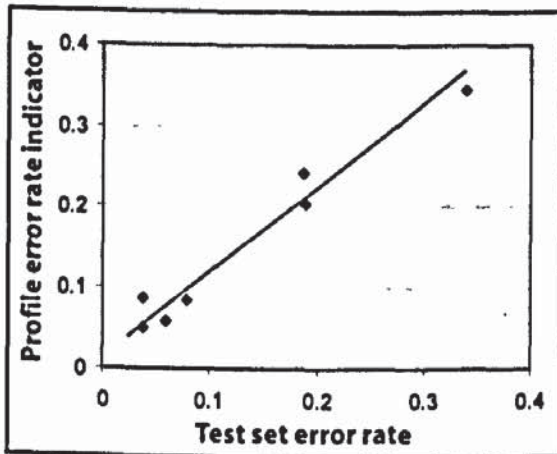


Figure 6.1: Error rate correlation

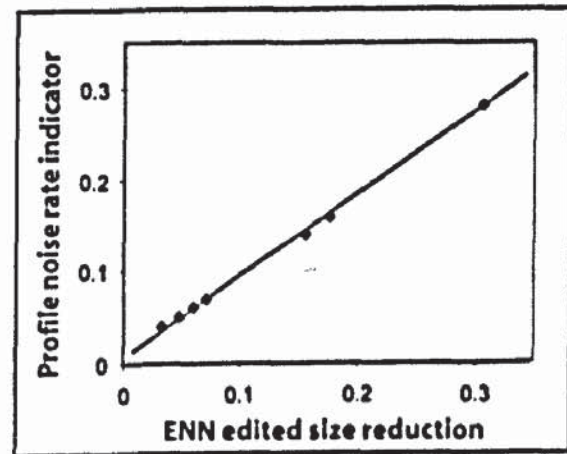


Figure 6.2: Noise level correlation

6.3 Complexity-Guided Case Discovery

In order to confirm that complexity-guided case discovery is useful we need to demonstrate that *useful* cases are discovered. In order to establish this our two complexity-guided algorithms have been compared with two benchmark algorithms. The test set accuracies achieved, on five UCI datasets, as different numbers of cases are discovered are recorded and the results from the four algorithms are compared.

6.3.1 Experimental Design

Four different case-discovery techniques have been implemented. All the algorithms identify two reference cases (a target case and its pair case) from within the case base. The

main difference between the four algorithms is in their approach to identifying these reference cases. Two of the algorithms are our complexity-guided algorithms described in Section 5.1.1 (COMPLEXITY and COMPLEXITY+) while the remaining two algorithms, which provide benchmarks for comparison, are as follows:-

- COMPETENCE uses competence-guided case discovery to create new cases between the nearest neighbour competence groups (McKenna & Smyth 2001a). Two reference cases are selected from different competence groups that are nearest to each other. The candidate case's feature values are set using the feature values of the reference cases' related sets. Figure 6.3 shows the algorithm operating on a representation of a small case base with cases belonging to two classes with a class boundary between them.

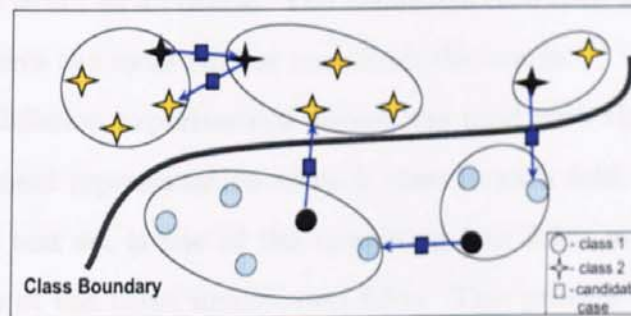


Figure 6.3: Illustration of COMPETENCE

- RANDOM is an uninformed algorithm that selects two reference cases at random from the case base and then uses these reference cases to create a candidate case in the same way as COMPLEXITY. This process continues until the required number of cases have been discovered.

Set-up of Experiment

Complexity-guided case discovery cannot guarantee that a valid case will be discovered. The objective is to supply a complete, candidate case to the case author to either accept or create a slight variation that corresponds to a valid case. This situation is difficult to replicate in an experimental evaluation because a domain expert is not available to validate the discovered cases. To simulate an expert our experimental design uses a pool

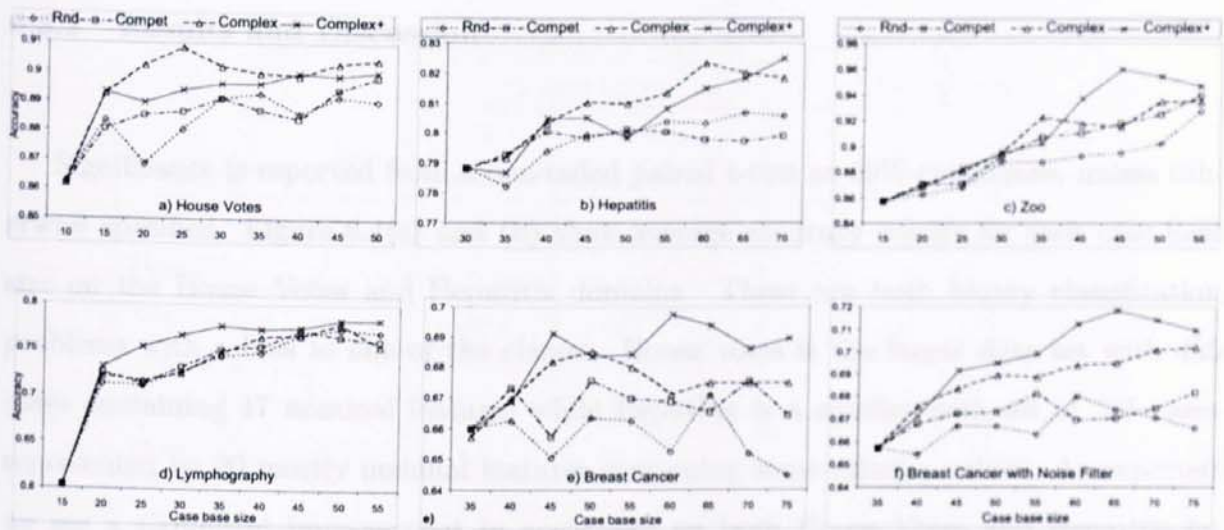


Figure 6.4: Accuracy of growing case bases as cases are discovered

of independent cases to act as an oracle. The candidate case then acts as a probe into this pool of cases to retrieve the most similar case from the oracle.

A 5-fold cross validation experimental design was used with the folds being stratified to ensure a proportional representation of each class in each fold. One fold was used as the training set, the test set is one of the remaining four folds in turn with the pool of cases being made up of the three unallocated folds. This process was repeated with the training set being allocated each of the 5 folds in turn resulting in 20 unique combinations of training set, test set and pool cases. There was no overlap between a training set and its associated test set and pool of cases.

The case base was initialised by randomly selecting a fixed number of cases from the training set. The starting size of the case base varied between 10 and 35 cases, depending on the dataset size and the difficulty of the problem. The algorithms were run on each trial on each dataset to discover between 5 and 40 cases in steps of 5. The results are plotted as a graph of the average accuracy for the test set for increasing case base size, as an increasing number of cases are discovered.

The experiments evaluate the effectiveness of the complexity-guided algorithms on test set accuracy with a varying number of cases being discovered. Test set accuracy is evaluated by 1-NN.

6.3.2 Results and Discussion

Significance is reported from a one-tailed paired t-test at 99% confidence, unless otherwise specified. Figure 6.4(a) and (b) show average accuracy results for each case base size on the House Votes and Hepatitis domains. These are both binary classification problems with a bias to one of the classes. House votes is the larger data set with 435 cases containing 17 nominal features while Hepatitis is a smaller data set of 155 cases represented by 20 mostly nominal features containing some missing values. As expected we see a significant improvement in accuracies on both House Votes and Hepatitis by the two complexity-guided algorithms (COMPLEXITY and COMPLEXITY+) over RANDOM and COMPETENCE. Perhaps surprisingly, the simpler COMPLEXITY gives the best performance on these datasets. This might be explained by these being binary problems with high accuracy suggesting a more simple boundary than on the other datasets. The simpler algorithm, by concentrating on only few areas of the problem space, appears to perform well on this type of domain.

Average accuracy for the Zoo and Lymphography domains appear in Figure 6.4(c) and (d). These are multi-class problems: Zoo has 101 cases split between 7 classes while Lymphography has 148 cases covering 4 classes. These domains have a similar number of features (18 and 19) with no missing values. Zoo contains only nominal features whereas Lymphography contains both nominal and numeric valued features. In both these domains COMPLEXITY+ produces the best performance with significant improvement over the other three algorithms. COMPLEXITY shows a significant improvement over RANDOM on the zoo domain but no difference over COMPETENCE. On Lymphography COMPLEXITY gave no improvement over either benchmark algorithm. The relatively poor performance of COMPLEXITY might be expected on these multi-class domains, as some of the classes contain a very small number of cases. In these situations COMPLEXITY will concentrate on providing cases to support the classes with low representation and provide insufficient support to the rest of the problem space. In contrast, COMPLEXITY+ uses clustering to provide a more balanced distribution of new cases.

Figure 6.4(e) shows average accuracy results on the Breast Cancer dataset. In Fig-

Table 6.3: Results summary according to significance.

Data Set	COMPLEXITY		COMPLEXITY+	
	vs. RANDOM	vs. COMPETENCE	vs. RANDOM	vs. COMPETENCE
House Votes	✓	✓	✓	✓
Hepatitis	✓	✓	✓	✓
Zoo	✓	no diff.	✓	✓
Lymphography	no diff.	no diff.	✓	✓
Breast Cancer	✓	✓ (95%)	✓	✓
Breast Cancer-Noise	✓	✓	✓	✓

ure 6.4(f) a noise filter, as described in Section 5.1.2, has been applied to all four algorithms for Breast Cancer. This is a binary classed domain with 9 multi-valued features containing missing data. The noise filter has been added because Breast Cancer is a more complex domain containing either noise or exceptional cases resulting in lower accuracies than the other domains. COMPLEXITY+ again produces the best performance with significant improvements over the other three algorithms. COMPLEXITY also shows a significant improvement over the two comparison algorithms in both experiments although the improvement over COMPETENCE without the noise filter is only significant at 95% confidence. The improved performance of COMPLEXITY+ over COMPLEXITY might again be explained by the simpler algorithm concentrating on supporting the noise or exceptional cases. It is interesting to see that, although the noise filter results in a small improvement in the performance of the benchmark algorithms it gives a large and significant improvement to the accuracies achieved by both the complexity-guided algorithms. This improvement is to be expected in noisy datasets because, by choosing cases on class boundaries, the complexity-guided algorithms will have a greater tendency to pick noisy cases.

Evaluation Summary

The results from the significance tests, comparing the two complexity-guided case discovery algorithms with the benchmark algorithms on each dataset, are summarised in Table 6.3. The first two columns display the improvement with COMPLEXITY while the

other two columns show significance results for **COMPLEXITY+**.

Overall **COMPLEXITY+**'s performance shows a significant improvement over the comparison algorithms on all the datasets and it provides the most consistent approach to case discovery of the algorithms studied. **COMPLEXITY** is shown to perform well on binary problems, particularly on simpler problems and on domains with low levels of noise, however, its performance on multi-class problems is only comparable with the benchmark algorithms.

The introduction of a noise filter stage gave significant accuracy improvements on the two complexity-guided discovery algorithms with Breast Cancer. This highlights the importance of noise filtering in noisy datasets.

6.3.3 Case Discovery with Noise Filtering

We have seen, in the previous section, that performance improved for the Breast Cancer dataset when a case validation stage was introduced, in which a discovered case is only accepted if its noise ratio lies below a predefined threshold. A conservative threshold of 1.5 was arbitrarily applied. It would be useful to establish whether this noise filter validation stage improves performance across all domains and to determine suitable threshold values. In this Section we evaluate the impact of applying a noise filter validation stage with different threshold levels to the complexity-guided case discovery algorithm.

Experimental Design

COMPLEXITY and **COMPLEXITY+** incorporating noise filtering have been applied to five UCI datasets with a range of different noise ratio thresholds. The 5-fold cross validation experiment, described in Section 6.3.1, is repeated with only **COMPLEXITY** and **COMPLEXITY+** for each of the five datasets. However, a post-processing noise filter validation stage is applied to both algorithms. The threshold at which the discovered case is excluded is varied from 0.75 to 1.5 in 0.25 steps. This allows the effect of aggressive and conservative noise filtering to be studied. The results from Section 6.3.2 (without noise filtering) provide a benchmark for comparison to evaluate whether noise filtering aids or hinders the case discovery process.

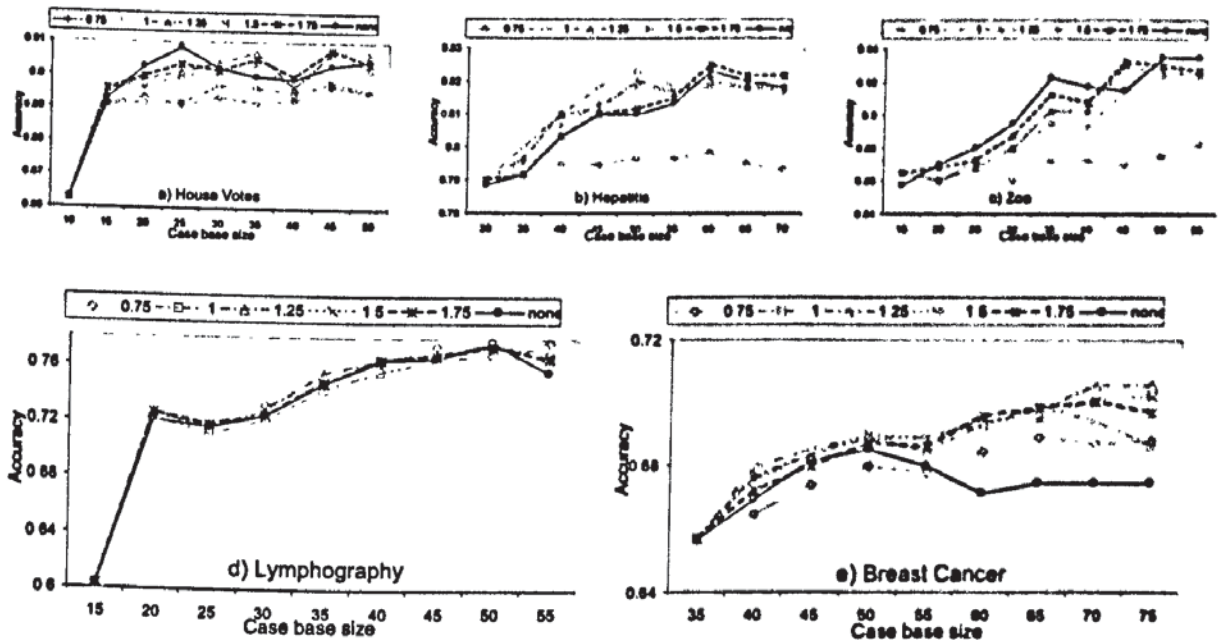


Figure 6.5: Accuracy as cases are discovered with COMPLEXITY with differing noise filtering levels

The aim of these experiments is to evaluate the effectiveness of applying noise filtering to the complexity-guided case discovery algorithms. Sample results are plotted as a graph, for each threshold, of the average accuracy for the test set for increasing case base size, as an increasing number of cases are discovered. Test set accuracy is again evaluated using 1-NN.

Results and Discussion

The average accuracy results for each case base size for the five datasets are shown graphically in Figure 6.5(a)-(e) for COMPLEXITY and Figure 6.6(a)-(c) for COMPLEXITY+. Results for each different filtering threshold are plotted as individual lines.

Due to the large number of results, rather than a detailed analysis for each algorithm, threshold and dataset, we make some general observations that apply across several datasets as follows:-

- Aggressive noise filtering (0.75) is almost always harmful. Even on noisy datasets, such as Breast Cancer, only small improvements in accuracy are achieved when compared to no filtering and only for larger case base sizes with COMPLEXITY+.

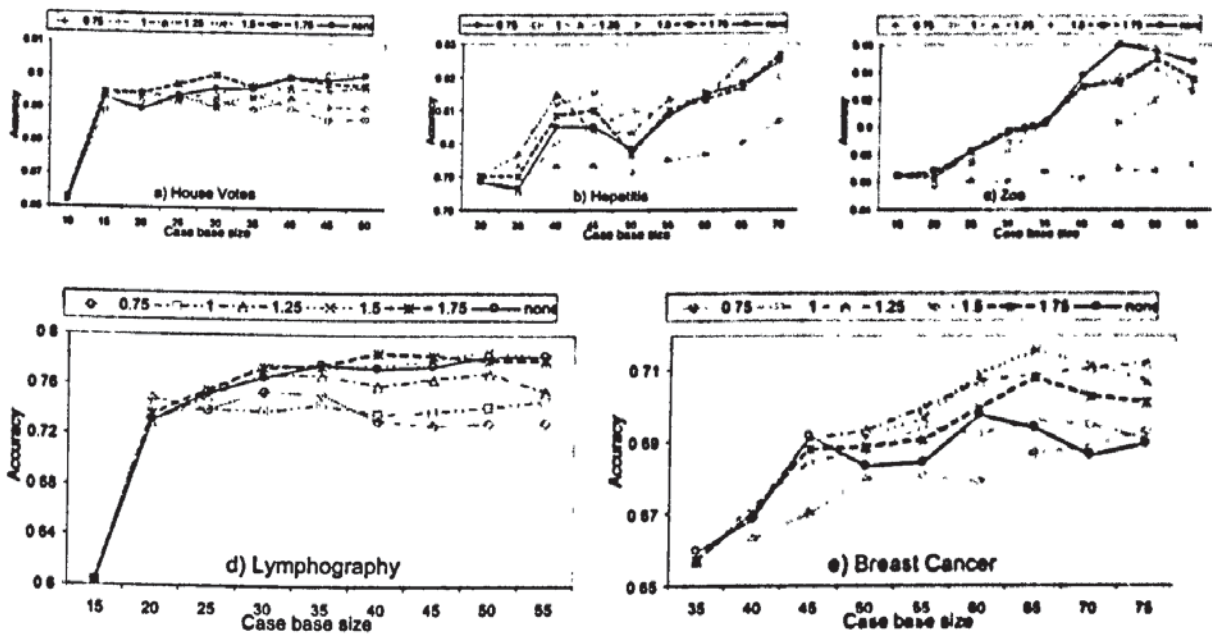


Figure 6.6: Accuracy as cases are discovered with COMPLEXITY+ for varying noise filtering levels

- Moderate levels of filtering, at 1.0 or 1.25 thresholds, generally give the best performance on the more noisy datasets such as Breast Cancer and Hepatitis.
- Conservative noise filtering (say at 1.75 or 1.5) provides a large improvement in test set accuracy when compared to no filtering on noisy datasets e.g. Breast Cancer. In general, conservative filtering is not very harmful even on data sets containing low levels of noise e.g. House Votes.
- Applying no filtering gives the best performance on datasets containing no noise.

Two approaches toward the noise filter validation stage would appear reasonable. One option is to apply conservative noise filtering at a threshold of 1.75 or 1.5 to the complexity algorithms in all situations with an expectation of *reasonable* performance. An alternative approach is to evaluate the noise characteristics of a dataset prior to applying the discovery algorithms and make an informed decision as to the level of filtering to apply. In datasets containing low levels of noise no filtering would be applied with conservative filtering being applied to datasets with moderate levels of noise and more aggressive filtering, using a threshold of 1 or 1.25, being applied to noisy datasets.

6.4 Complexity-Guided Editing

We evaluate our complexity-guided approach to redundancy removal from two perspectives. Firstly, to confirm our hypothesis that setting an editing threshold of zero will result in an edited case base with little or no loss in competence. Secondly, to compare the performance of our complexity-guided approach with several existing redundancy reduction algorithms. The algorithms being compared can be split into three categories.

- Complexity Threshold Editing (CTE) is our new redundancy reduction algorithm described in Section 5.2.2 and is directly comparable with existing redundancy reduction algorithms. It is evaluated with four different complexity thresholds: 0, 0.1, 0.2 and 0.3. The zero threshold gives conservative editing with the level of editing gradually increasing as the threshold increases. Thresholds of 0.1 and 0.2 give a moderate level of editing while the 0.3 threshold provides an aggressive editing approach.
- Existing redundancy reduction algorithms: the three algorithms chosen are all based on the competence model developed by Smyth & McKenna's (1998) and use coverage and reachability sets to identify *important* cases. These are modern redundancy reduction algorithms that aim to reduce the size of the case base while maintaining competence. They are described in more detail in Section 2.2.1 and have been shown to perform well in previous comparisons (Delany & Cunningham 2004, Brighton & Mellish 2002, McKenna & Smyth 2000). Each algorithm provides a different balance between compaction of the case base and competence:
 - Conservative Redundancy Reduction (CRR) provides a conservative approach to redundancy editing. With Delany & Cunningham's (2004) algorithm a case with the smallest coverage set is selected from the case base for addition to an edited set first, and any cases that it solves are deleted from the case base. The process is repeated until no more cases remain in the case base.
 - Iterative Case Filtering (ICF) falls in the middle giving a moderate level of case base compaction. With Brighton & Mellish's (2001) editing algorithm a case is deleted if its reachability set is larger than its coverage set, i.e. more cases can

solve the case than it can solve itself. The process is repeated until no more cases are removed. This results in boundary cases being retained and central cases being removed. Wilson & Martinez's (1997) RT3 algorithm would also have been a suitable algorithm for moderate levels of case base compaction. However, as previous evaluations show similar performance to ICF only one algorithm was chosen (McKenna & Smyth 2000).

- Relative Cover Editing (RC) is an aggressive algorithm deleting the highest number of cases. Smyth & McKenna's (1999a) competence-guided editing techniques use local case information from their competence model to rank cases prior to case selection using Hart's (1968) Condensed Nearest Neighbour rule, so that redundant cases are presented later in the editing process. Several ranking measures are proposed based on a case's *coverage* and *reachability sets*. We use the *relative cover* ranking (RC), which is shown to give a large reduction in case base size while retaining competence.
- Noise reduction algorithms (ENN and RENN). These algorithm, also described in Section 2.2.1, aim to improve competence but remove only a few cases and are not directly comparable with redundancy reduction algorithms. They are included in the evaluation because RENN has been used as the pre-processing algorithm for all the redundancy reduction algorithms including CTE. RENN provides a benchmark for accuracy that the redundancy reduction algorithms aim to maintain.

6.4.1 Experimental Design

A ten times 10-fold cross validation experimental set-up is used giving 100 case base/test set combinations per experiment. The editing algorithms were applied to each case base and the resulting edited set size is recorded. Test set accuracy, using 1-NN retrieval, was measured for the original unedited case base and for each of the edited sets created by the editing algorithms.

Comparisons have been made on seven UCI datasets and the averaged results are shown in Table 6.4 and 6.5. Table 6.4 contains the average test set accuracy for the unedited dataset and for each editing algorithm on each domain. The highest accuracy

result achieved by the redundancy reduction algorithms in each domain is highlighted in bold. Table 6.5 gives the unedited dataset size in column 2 together with the edited dataset size as a proportion of the original in the other columns. The values in bold are the size reduction achieved by the redundancy algorithm with the highest accuracy. Both tables include an *average* row which should be used with care as it is calculated across different domains.

Table 6.4: Comparison of average test set accuracy for alternative editing algorithms

CASE BASE	ORIG	REDUNDANCY			CTE				NOISE	
		CRR	ICF	RC	0	0.1	0.2	0.3	RENN	ENN
Breast Cancer	0.661	0.740	0.736	0.688	0.738	0.734	0.728	0.709	0.753	0.736
Hepatitis	0.808	0.839	0.833	0.821	0.850	0.853	0.847	0.822	0.834	0.862
House Votes	0.922	0.905	0.901	0.904	0.922	0.916	0.898	0.854	0.911	0.920
Iris	0.940	0.947	0.931	0.943	0.940	0.933	0.882	0.878	0.952	0.952
Lymphography	0.812	0.759	0.749	0.757	0.775	0.775	0.776	0.758	0.772	0.781
Wine	0.963	0.957	0.934	0.923	0.950	0.924	0.884	0.822	0.948	0.953
Zoo	0.957	0.906	0.902	0.904	0.921	0.901	0.876	0.778	0.904	0.926
Average	0.866	0.865	0.855	0.850	0.875	0.862	0.842	0.803	0.868	0.876

Table 6.5: Comparison of edited case base size for alternative editing algorithms

CASE BASE	ORIG	REDUNDANCY			CTE				NOISE	
		CRR	ICF	RC	0	0.1	0.2	0.3	RENN	ENN
Breast Cancer	258	0.248	0.160	0.071	0.604	0.450	0.292	0.163	0.674	0.694
Hepatitis	140	0.403	0.082	0.061	0.355	0.265	0.186	0.102	0.706	0.824
House Votes	392	0.471	0.035	0.038	0.155	0.093	0.061	0.038	0.908	0.928
Iris	135	0.389	0.296	0.065	0.177	0.078	0.038	0.037	0.952	0.952
Lymphography	134	0.415	0.180	0.152	0.625	0.460	0.322	0.189	0.815	0.846
Wine	161	0.439	0.159	0.099	0.208	0.098	0.056	0.033	0.965	0.967
Zoo	91	0.355	0.486	0.110	0.241	0.138	0.096	0.075	0.927	0.938
Average	187	0.394	0.153	0.074	0.328	0.221	0.146	0.087	0.853	0.870

6.4.2 Results and Discussion

The results of the evaluation can be summarised in each of the categories as follows:

- The CTE algorithm provides the highest accuracy of the redundancy reduction algorithms in six of the seven domains. At zero complexity threshold, CTE has the highest average accuracy of 87.5% compared to 86.5% for CRR. This is achieved with smaller case base sizes, 32.8% of original size on average compared to 39.4%, showing that CTE is an excellent algorithm for conservative redundancy reduction. At moderate levels of redundancy reduction, with a threshold of 0.1, CTE achieves slightly better accuracies than ICF but retains slightly more cases. Overall the performance is comparable with ICF. With higher complexity thresholds, for aggressive redundancy reduction, CTE does not perform so well and is outperformed by RC.
- The three existing redundancy reduction algorithms all provide a different compromise on the trade-off between case base compaction and maintaining competence. CRR, designed to take a conservative approach to redundancy reduction, has the highest accuracy on each domain and the highest average accuracy of 86.5% compared to 85.5% for ICF and 85.0% for RC. However, CRR obtains the improved accuracy by retaining, on average, 39% of the cases, more than twice that of ICF (15%) and five times RC (7%). Very aggressive redundancy reduction is achieved by RC but the results confirm that this is at the expense of loss of accuracy. The performance of ICF lies between the others on both competence retention and case base size reduction.
- There is little to choose between the performance of the noise reduction algorithms. In these datasets ENN gives the highest average accuracy but that is probably because many of these datasets are not noisy and ENN gives the best results on data with low levels of noise. RENN removes slightly more cases and generally performs better on noisy data but worse on low noise datasets. It is worth noting that on four of the datasets both the noise reduction algorithms harm accuracy but on Breast Cancer and Hepatitis substantial accuracy gains are achieved by noise reduction.

CTE provides the best performance for conservative redundancy reduction, providing superior accuracy on six out of the seven domains. We checked the significance of these differences using a 2-tailed t-test with 95% confidence level. The superiority of CTE was found to be significant in 4 domains; Hepatitis, House Votes, Lymphography and Zoo.

As expected, setting a zero level threshold maintained accuracy at a similar level to that achieved after RENN noise reduction in all the domains and overall there was actually a slight increase in accuracy from 86.8% to 87.5%. This confirms that at the local level the case complexity measure identifies redundant cases and at a global level the redundancy indicator estimated from the complexity profile is a good predictor of the level of redundancy within a case base. When the complexity threshold is increased above zero, accuracy initially falls away gradually at first, as non-redundant cases start to be deleted and then more quickly as cases nearer to decision boundaries are deleted.

The performance of CTE for aggressive levels of redundancy reduction with the higher complexity thresholds was disappointing. This suggests that while case complexity provides a good measure for identifying redundant cases away from boundaries, it is not so good at selecting between alternative boundary cases.

The expectation that accuracy would fall as the size of the edited case base falls is corroborated both for the existing redundancy reduction algorithms and for CTE. This confirms previous research results (Delany & Cunningham 2004) that there is a trade-off between the conflicting objectives of compaction of the case base and maintaining competence.

The inconsistent performance of the noise removal algorithms across the different datasets highlights the need to apply different maintenance strategies for different domains. Complexity profiling of the case base can play a role in identifying appropriate strategies for a case base.

6.5 Error Reduction

In order to demonstrate that TER can improve accuracy we evaluate its' performance against several existing noise reduction algorithms. The algorithms are evaluated in two stages in this section. In our initial experiments we apply the algorithms to existing UCI datasets and compare accuracy and size reduction results achieved. Then in the second stage of the evaluation we artificially introduce higher levels of noise into the datasets to examine the algorithms performance in more challenging environments. TER is compared with two classic benchmark noise reduction algorithms: Wilson Editing (ENN) and Re-

peated Wilson Editing (RENN). The benchmark algorithms are described in Section 2.2.1.

6.5.1 Initial Experiments

A ten-times 10-fold cross-validation experimental set-up is used giving one hundred case base/test set combinations per experiment. The editing algorithms were applied to each case base and the resulting edited set size recorded. Test set accuracy, using 1-NN retrieval, was measured for the original case base and for each of the edited sets formed by the editing algorithms.

Comparisons have been made on seven UCI datasets. Table 6.6 contains the average test set accuracy for each algorithm on each domain. Table 6.7 gives the unedited case base size in column 2 together with the edited case base size as a proportion of the original in the other columns. In both tables the editing algorithm that achieved the highest accuracy in each domain is highlighted in bold.

Table 6.6: Comparison of average test set accuracy

Case Base	ORIG	ENN	RENN	TER
Breast Cancer	0.661	0.746	0.753	0.758
Hepatitis	0.808	0.826	0.827	0.837
House Votes	0.921	0.919	0.911	0.924
Iris	0.940	0.951	0.951	0.955
Lymphography	0.812	0.777	0.765	0.798
Wine	0.965	0.954	0.948	0.965
Zoo	0.957	0.919	0.895	0.946

TER provides the highest accuracy in all seven domains. We checked the significance of these differences using a 2-tailed t-test with 95% confidence level. The superiority of TER was found to be significant in 4 domains: Hepatitis, Lymphography, Wine and Zoo. TER achieves its performance gain by using the stopping criteria to vary the level of editing at the decision boundaries. In some domains, where smoothing the decision boundary is found to improve accuracy, TER removes far more cases than the benchmark algorithms e.g. Hepatitis. In other domains, where boundary smoothing is found to be harmful, TER removes less cases than the benchmarks, for example in Wine no cases are removed at all. It is worth noting that in two domains the original accuracy was higher than any of the

Table 6.7: Comparison of edited case base size

Case Base	ORIG	ENN	RENN	TER
Breast Cancer	258	0.69	0.67	0.67
Hepatitis	140	0.82	0.80	0.70
House Votes	392	0.93	0.91	0.97
Iris	135	0.95	0.95	0.93
Lymphography	134	0.84	0.81	0.76
Wine	161	0.97	0.97	1.00
Zoo	91	0.94	0.93	0.98

editing algorithms. In these datasets any editing appears harmful although TER appears least harmful. In a comparison of the benchmark algorithms RENN removes more cases but is slightly outperformed by ENN which achieves higher accuracies in four domains compared to two for RENN .

6.5.2 Experiments on Datasets with Artificial Noise

The same experimental set-up used for the initial experiments was adopted for a second set of experiments with the exception that differing levels of noise were artificially introduced into the case base. Noise was introduced by randomly selecting a fixed proportion of the cases in the case base and changing the class of their solution. The algorithms were evaluated after the introduction of 10%, 20% and 30% noise levels. Table 6.8 shows the average test set accuracy for each algorithm on Breast Cancer, Hepatitis and Lymphography with 10%, 20% and 30% noise introduced. Table 6.9 displays the unedited case base size in column 2 and the edited case base size as a proportion of the original, for the relevant dataset and noise level, in the remaining columns. Again, the algorithm that achieved the highest accuracy for each domain and noise level is highlighted in bold and also in italics if it significantly outperformed the other algorithms.

As expected the accuracy on the original case base falls dramatically with increasing noise levels. All the noise reduction algorithms help slow the degradation in accuracy and, unlike our initial experiment, they dramatically improve on the accuracy achieved with the unedited case base. Overall TER gives the strongest performance, recording the highest accuracy in 6 of the 9 experiments. However, the improvement is only significant

Table 6.8: Comparison of average test set accuracy

Case Base	ORIG	ENN	RENN	TER
Breast Cancer (10%)	0.631	0.708	0.729	0.735
Breast Cancer (20%)	0.605	0.677	0.697	0.696
Breast Cancer (30%)	0.583	0.646	<i>0.679</i>	0.655
Hepatitis (10%)	0.744	0.833	0.829	0.830
Hepatitis (20%)	0.708	0.816	0.810	0.817
Hepatitis (30%)	0.663	0.783	<i>0.809</i>	0.792
Lymphography (10%)	0.753	0.762	0.758	<i>0.789</i>
Lymphography (20%)	0.713	0.734	0.731	<i>0.762</i>
Lymphography (30%)	0.641	0.688	0.718	0.724

in 2 experiments (Lymphography 10% & 20%) and RENN gives the highest accuracy in the remaining three experiments. RENN is particularly strong with data containing a high proportion of noise. It would appear that TER's competitive advantage gained by smoothing the boundary regions between classes is diminished in data containing high levels of noise, possibly because the noise creates false decision boundaries that the algorithm attempts to maintain.

Table 6.9: Comparison of edited case base size

Case Base	ORIG	ENN	RENN	TER
Breast Cancer (10%)	258	0.63	0.61	0.60
Breast Cancer (20%)	258	0.59	0.54	0.65
Breast Cancer (30%)	258	0.56	<i>0.50</i>	0.63
Hepatitis (10%)	392	0.73	0.71	0.63
Hepatitis (20%)	392	0.66	0.62	0.56
Hepatitis (30%)	392	0.60	<i>0.53</i>	0.47
Lymphography (10%)	134	0.74	0.69	<i>0.55</i>
Lymphography (20%)	134	0.65	0.60	<i>0.49</i>
Lymphography (30%)	134	0.58	0.50	0.44

6.6 Chapter Summary

The global indicators extracted from a case base's profile of case complexities give a measure of the level of complexity, redundancy and noise inherent in the data. These

indicators are shown to be good predictors of the real values inherent in the data by the close correlation with alternative experimental measures.

The effectiveness of **COMPLEXITY** and **COMPLEXITY+**, our new case discovery algorithms, were demonstrated on 5 public domain datasets. In general, a significant improvement in test accuracy was observed with these new techniques compared to the random and competence-guided algorithms used as benchmarks. **COMPLEXITY** performed well on simple binary domains but suffered on multi-class problems or on datasets containing noise. **COMPLEXITY+**, which incorporated a clustering stage, provided the most consistent performance across the range of datasets. A conservative noise filter stage was found to enhance the performance of **COMPLEXITY** and **COMPLEXITY+** on noisy datasets.

The effectiveness of the new redundancy removal algorithm, **CTE**, was evaluated on seven UCI datasets. The algorithm was shown to provide superior performance characteristics when compared to existing techniques for conservative levels of editing and comparable performance at moderate levels of editing. One limitation of the approach is an average performance for aggressive editing because the complexity measure does not make a balanced selection between alternative boundary cases. Enhancements are being investigated to improve this selection on boundaries.

In general, **TER** is shown to provide superior error reduction performance when compared to benchmark techniques for case bases containing low and medium levels of noise. One limitation of the approach is its ability to identify harmful cases when the case base contains high levels of noise and boundaries become difficult to identify. The results confirm that error reduction can harm performance in dataset with low levels noise. Careful consideration should be given to the domain and the structure of the case base to ensure there is a need for noise reduction before removing case knowledge with an editing algorithm. The complexity and F:E profile provides a tool for the knowledge engineer to make an informed decisions on the need for error reduction maintenance.

Chapter 7

Conclusions and Future Work

This thesis has investigated some of the theoretical and practical issues surrounding modelling and maintenance of the case knowledge used in CBR systems. This chapter concludes the thesis with a discussion of the contributions made, future directions and lessons learned.

7.1 Achievements and Contributions

In this section we look at the contributions of the work by revisiting the initial project objectives, described in Section 1.3, and considering the extent to which they have been achieved.

1. **Develop a technique to model the problem-solving capabilities of a case base.**

This research has highlighted the importance of modelling CBR competence in order to reduce the need for case base evaluation experiments and to assist in the development of case base maintenance algorithms. Experiments have identified classification datasets in which the current models do not give a good correlation to competence. Three reasons for this lack of correlation have been identified: the inclusion of redundant cases is not adequately reflected, statistical measures such as group case density do not give a good measure of competence, and, most importantly, problem complexity is not considered.

CBR is an effective problem-solving methodology in domains where similar problems have similar solutions. The foundation of our approach is a complexity measure that tests the extent to which this axiom holds true at a local level. This is accomplished by making the assumption that the case base itself is representative of problems the system will face. First the neighbours of each case (i.e. similar problems) are identified and then the mix of solutions within this group of similar problems is compared by way of our complexity measure.

A ranked profile of case complexities has been used to give the knowledge engineer a global view of the mix of complexities within the problem space. We have also shown how the complexity profile can provide a prediction of the level of complexity, redundancy and noise inherent in the data that correlates well with the real values. In the first instance the complexity profile gives an indication of the suitability of a domain for problem-solving using the CBR methodology i.e. do similar problems have similar solutions? The interpretation of several profiles has been discussed to show how they can aid the knowledge engineer develop suitable case base maintenance policies. Profiling also provides the opportunity to create a benchmark for comparison with future versions of the case base to monitor the impact of changes over time.

2. Develop a technique to identify gaps and create new cases to fill them.

In a novel case discovery approach the complexity measure, in conjunction with a case's nearest unlike neighbour, guides the case discovery process by identifying areas of the problem space in which the system is unsure of the solution. The idea of placing new cases on classification boundaries appears to be intuitively sensible in that it mirrors the approach of recently developed case base editing algorithms. **COMPLEXITY** and **COMPLEXITY+**, two new complexity-guided algorithms, are introduced and their effectiveness demonstrated on public domain datasets. In general, a significant improvement in test accuracy is observed with these new techniques compared to the random and competence-guided algorithms used as benchmarks. **COMPLEXITY** performs well on simple binary domains but suffers on multi-class problems or on datasets containing noise. **COMPLEXITY+**, which incorporates a clustering stage,

provides the most consistent performance across the range of datasets. A noise filter stage was found to enhance the performance of **COMPLEXITY** and **COMPLEXITY+** on noisy datasets.

3. Develop a Case Base Maintenance algorithm that identifies redundant cases.

The contribution from this area of the research is in the use of the local case complexity measure to identify redundant cases located in areas of the problem space in which the mix of solutions show a strong coherence. In classification tasks these areas are typified by single class clusters. The redundant cases are identified by applying a threshold to the complexity profile and editing cases with complexity values on or below the threshold. The approach provides the knowledge engineer with an element of control over the compromise required between the contradictory objectives of the reduction in case base size and the retention of competence.

Several refinements that alleviate problems associated with mutual redundancy and noise are incorporated into a new redundancy removal algorithm: **Complexity Threshold Editing**. The algorithm is shown to provide superior performance characteristics when compared to existing techniques for conservative levels of editing and comparable performance at moderate levels of editing.

4. Develop a Case Base Maintenance algorithm that identifies harmful cases.

This objective is achieved with the aid of the **F:E** ratio calculated for each case. This local case distance ratio gives a measure of a case's position in relation to neighbours of its own class and neighbours with a different class. Following a similar approach to the one adopted for complexity profiling, a ranked profile of the **F:E** ratios is plotted and used to guide the editing process. In contrast to the redundancy removal approach, harmful cases are identified by applying a threshold to the ratio profile and removing cases with ratios above the threshold.

The algorithm focuses on deleting both noisy cases and harmful cases from boundary regions to give smoother decision boundaries between classes. A stopping criteria is introduced to ensure that the level of smoothing is adjusted to suit the domain.

We have introduced TER and demonstrated its effectiveness on UCI datasets. In general, TER provides superior performance characteristics when compared to benchmark techniques for case bases containing low and medium levels of noise.

Noise reduction can also harm performance. Careful consideration should be given to the domain and the structure of the case base to ensure there is a need for noise reduction before removing case knowledge with an editing algorithm. The F:E ratio profile provides a tool for the knowledge engineer to make an informed decision on the need for case base maintenance to remove noise.

5. Create a visualisation tool that demonstrates case coverage and allows the user to view redundancy and gaps.

A novel set of interfaces have been designed as the front end of an integrated case base maintenance tool. The interfaces display the complexity profile to provide a graphical view of the mix of complexities in the case base. This gives a measure of the complexity of the problem, the level of redundancy and the level of noise within the data.

A two-dimensional spring based visualisation of the case base gives a *map* of the case base displaying local complexities as a colour gradient applied to the individual cases.

The interactive prototype developed to demonstrate our approach to case base maintenance gives an explanation of the maintenance process by highlighting the differences between the original and proposed case bases on the visualisations.

7.2 Going Forward

In common with most research activity, as this work has progressed potential new areas of research have been identified. In this section we look at some of the limitations of our work in conjunction with possible extensions or future work. The limitations take two forms: as a result of shortcomings in the approaches themselves, and through restrictions placed on the scope of the work

7.2.1 Shortcomings in Approaches

With hindsight some of the algorithms introduced in this thesis may have been developed differently or, at least, a few alternative approaches or extensions would have been considered and investigated. Here we identify three such alternatives that provide potential to improve the performance of our algorithms in future work.

- The complexity measure chosen looks at the mix of solutions in local areas of the problem space to identify areas of uncertainty. The measure used is calculated for each case and considers the proportion of the case's neighbours belonging to the same class as itself giving a strong weighting to the nearer neighbours. The weighting is applied in relation to a case's position in a ranked list but does not consider the actual distance to the neighbour from the case under consideration. A more complex but finer grained, alternative approach would be to apply a weighting in relation to the distance between a case and its neighbours. This could be considered similar to using a weighted majority vote with k -NN. A comparison between these alternative approaches would be interesting and could identify the benefits of each.
- The complexity-guided redundancy reduction algorithms have exhibited excellent performance for conservative redundancy reduction tasks. However, one limitation of the approach is an average performance for aggressive editing when applying higher valued thresholds. This is because using the complexity measure alone to guide editing does not result in a balanced selection between alternative boundary cases. Rather the retained cases are concentrated in only a few *complex* areas of the problem space. Adopting an approach in which the cases are formed into clusters, prior to the use of the complexity measure to guide the case editing selection from each cluster, may give a more even distribution of retained cases along the decision boundaries.
- One limitation of the complexity-guided case discovery algorithms is that they restrict their search space to finding new cases within the problem space already identified by existing cases i.e. new cases are discovered between two existing cases. Future work may look at developing complimentary approaches for the very early growth

stages of a case base, perhaps by using domain knowledge to seed the case base.

7.2.2 Limitations of Scope

Our work has focused on importance of decision boundaries and on the variance in solutions present within local areas of the problem space. In taking this approach the scope of our research has been limited in several ways resulting in the potential for extensions that expand the scope of the work. For example, only lazy learner classifiers, such as k -NN, have been considered and it would be interesting to investigate whether other classifiers, e.g. support vector machines, could benefit from a similar editing approach. However, two key directions would seem to be particularly promising and exciting.

Beyond Classification Problems

In classification tasks it is relatively easy to measure the local mix of solutions by counting the number of cases belonging to each class and, as we have discussed above, the approach could be extended to consider the distance between cases. However in unsupervised tasks, in which cases are not assigned class labels, it may still be possible to measure the local mix or variance in solutions. In order to extend our approach beyond classification problems we need to be able to measure the similarity between solutions. In some tasks this is achievable, for example, in textual CBR both the problem and the solution are often in textual form. If the similarity can be measured between the problem part of a case it should be equally possible to measure the similarity between solutions.

Once the similarity between solutions has been identified, the coherence among the solutions could be measured, at a local level, either independently or in relation to the similarity between the problems. A measure of solution coherence would allow the extent to which similar problems have similar solutions to be gauged, thus, providing an indication of the suitability of a domain for problem-solving by CBR. By adopting similar approaches as used here for classification problems, ranked profiles of the case base could give a global view of the case base and inform specific maintenance algorithms.

Beyond Case Base Maintenance

In this research we have developed a case base model based on a case complexity measure that provides information about local areas of the problem space. The local information has been used to maintain a CBR system's case knowledge. The variation in solutions in local areas of the problem space could be used to maintain the other CBR knowledge containers e.g. retrieval knowledge or vocabulary.

- Retrieval knowledge is used to identify similar cases to a target problem. In CBR this often involves setting the relative importance of each attribute by means of an attribute weight. Suitable attribute weights are typically determined by statistical or evolutionary approaches and applied either globally in all situations or in relation to each class. A measure of the local uncertainty, by way of the local mix of solutions, gives the opportunity to apply a finer grained approach. The relative importance of attributes could be determined such that class separation is maximised at a local level with attribute weights being applied locally and associated with either individual cases or clusters of cases.
- Vocabulary refers to the way in which a case is represented and in CBR often involves selecting a suitable set of attributes. Because irrelevant attributes make classification more difficult and computationally expensive, maintenance to this knowledge container usually involves implementing filter or wrapper attribute selection techniques to identify a subset of *discriminating* attributes. A single representation is normally used across the whole problem space, however, in an approach similar to that discussed above for identifying local attribute weights, information about local areas of the problem space could be used to inform a search for a local set of discriminating attributes.

Knowledge of local areas of the problem space can be used to inform a broader range of CBR maintenance tasks than just case base maintenance as considered in this research. The complexity measure, giving a local measure of uncertainty, could potentially even be used outside purely maintenance tasks, for instance to give a measure of solution confidence or as an evaluation measure of alternative system designs.

7.3 Final Comments

Finding a time at which to end a piece of work is never easy because research tends to be a continuous process rather than independent pieces of work. However the work reported here provides a good place to draw breath and conclude the thesis. It provides a comprehensive look at one type of CBR task (i.e. classification) and at the maintenance of one of CBR's knowledge containers (i.e. case knowledge).

The main contribution of the work is in identifying boundary regions as critical areas of the problem space for determining the performance of CBR classification systems and showing how these region can be found by looking at the local mix of solutions. By looking at the relationship between the similarity among problems and the similarity among solutions independently we are able to measure the complexity within the case base and gauge its suitability for problem-solving by the CBR methodology i.e. do similar problems have similar solutions? The local information has been made available to the knowledge engineer graphically, reducing the need for experimental evaluation and has been used to inform maintenance algorithms to perform typical maintenance tasks: case discovery, redundancy reduction, and error reduction.

There has been considerable research effort applied to case base maintenance for classification tasks over several decades. Most of the approaches have delivered algorithms that perform only part of the overall maintenance task. This work has applied a consistent approach across the range of typical maintenance tasks to give a set of integrated techniques. It is hoped that in applying a consistent approach the overall contribution is greater than a sum of the individual parts.

The performance of an algorithm is often not the only criterion on which it will be judged as other, often less obvious factors, may affect the acceptance of an algorithm in real situations. Explanation and control are two central themes that apply throughout our work that should assist their transition or acceptance as suitable maintenance techniques for commercial systems.

- Explanation

A knowledge engineer gains confidence in a maintenance approach that provides strong performance. However, confidence is also improved in approaches where the

process is transparent and deficiencies can be identified and resolved. Explanation of the maintenance process and result should be a key criterion in the development of CBR knowledge maintenance techniques. We have applied a consistent and transparent approach to case base maintenance throughout this research and developed a prototype that demonstrates how visualisations can be used to assist in the provision of explanations to the user.

- Control

Case base maintenance is a balance between competing factors. We have developed an interactive approach in which the knowledge engineer retains control so that informed decisions can be made that consider both the impact of the maintenance algorithm and the system objectives. Leaving control of the maintenance process with the knowledge engineer is likely to aid the acceptance of maintenance as a crucial aspect of CBR system development for commercial systems.

Two promising areas for future work have been identified. First, extending the approach beyond classification problems to consider unsupervised tasks in which the similarity between solutions can be identified. This is a hot topic, particularly in textual CBR where large case bases are common and few reliable case base maintenance techniques have been developed. Second, extending the approach to maintain the other CBR knowledge containers, in particular the similarity knowledge, would appear to have great potential.

Bibliography

- Aamodt, A. & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches, *A I Communications* 7(1): 39-52.
- Ackoff, R. & Sasieni, M. (1968). *Fundamentals of Operations Research*, John Wiley & Sons, New York.
- Aha, D. (1997). Special issue on lazy learning, *Artificial Intelligence Review* 11: 7-10.
- Aha, D., Kibler, D. & Albert, M. (1991). Instance-based learning algorithms, *Machine Learning* 6(1): 37-66.
- Ankerst, M., Berchtold, S. & Keim, D. A. (1998). Similarity clustering of dimensions for an enhanced visualization of multidimensional data, *In Proceedings of IEEE Symposium on Information Visualization, (InfoVis'98)*, IEEE Computer Society Press, pp. 52-60.
- Barletta, R. (1994). A hybrid indexing and retrieval strategy for advisory CBR systems built with remind, *In Proceedings of the Second European Workshop on Case-Based Reasoning*, pp. 49-58.
- Blake, C., Keogh, E. & Merz, C. (1998). UCI repository of machine learning databases.
*<http://www.ics.uci.edu/~mllearn/MLRepository.html>
- Brighton, H. & Mellish, C. (1999). On the consistency of information filters for lazy learning algorithms, in J.M.Zytkow & J.Rauch (eds), *In Proceedings of Principles of Data Mining and Knowledge Discovery: 3rd European Conference*, LNAI 1704, Springer, Prague, Czech Republic, pp. 283-288.

- Brighton, H. & Mellish, C. (2001). Identifying competence-critical instances for instance-based learners, in H. Motoda & H. Liu (eds), *Instance Selection and Construction for Data Mining*, Kluwer Academic Publishers, pp. 77-94.
- Brighton, H. & Mellish, C. (2002). Advances in instance selection for instance-based learning algorithms, *Data Mining and Knowledge Discovery* 6(2): 153-172.
- Brodley, C. (1993). Addressing the selective superiority problem: Automatic algorithm/mode class selection, *Machine Learning: Proceedings of the Tenth International Conference*, pp. 17-24.
- Brodley, C. & Friedl, M. A. (1996). Identifying and eliminating mislabeled training instances, *In Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 799-805.
- Cameron-Jones, R. (1992). Minimum description length instance-based learning, *In Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence*, pp. 368-373.
- Chang, C. (1974). Finding prototypes for nearest neighbour classifiers, *In IEEE Transactions on Computers*, Vol. C-23, pp. 1179-1184.
- Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* 13(1): 21 - 27.
- Craw, S., Jarmulak, J. & Rowe, R. (2001). Maintaining retrieval knowledge in a case-base reasoning system, *Computational Intelligence* 17(2): 346-363.
- Cunningham, P. (1998). CBR: Strengths and weaknesses, *In lecture notes in artificial intelligence 1416 (Vol. 2)*, pp. 517-523.
- Delany, S. J. & Cunningham, P. (2004). An analysis of case-base editing in a spam filtering system, *Proceedings of the 7th European Conference on Case-Based Reasoning*, LNAI 3155, Springer, Madrid, Spain, pp. 128-141.
- Domingos, P. (1995). Rule induction and instance-based learning: A unified approach,

-
- Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJ-CAI 95)*, pp. 1226–1232.
- Doyle, D., Cunningham, P., Bridge, D. G. & Rahman, Y. (2004). Explanation oriented retrieval, *Proceedings of the 7th European Conference on Case-Based Reasoning*, LNAI 3155, Springer, Madrid, Spain, pp. 157–168.
- Eades, P. (1984). A heuristic for graph drawing, *Congressus Numerantium* 42: 149–160.
- Falkman, G. (2002). The use of a uniform declarative model in 3D visualisation for case-based reasoning, *Proceedings of the 6th European Conference on Case-Based Reasoning*, LNAI 2416, Springer, pp. 103–117.
- Francis, A. & Ram, A. (1993). Computational models of the utility problem and their application to a utility analysis of case-based reasoning, *Proceedings of the Workshop on Knowledge Compilation and Speed-Up Learning*.
- Gates, G. (1972). The reduced nearest neighbor rule, *IEEE Transactions on Information Theory* 18(3): 431–433.
- Giarratano, J. & Riley, G. (1994). *Expert Systems: Principles and Programming*, PWS Publishing Co, Boston, MA.
- Gomes, Pereira, Carreiro, Paiva, Seco, Ferreira & Bento (2003). Evaluation of case-based maintenance strategies in software design, *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 03)*, LNAI 2689, Springer, pp. 186–200.
- Gonzalez, A. & Dankel, D. (1993). *The Engineering of Knowledge-Based Systems: Theory and Practice*, Prentice-Hall, Englewood Cliffs, CA.
- Hart, P. (1968). The condensed nearest neighbour rule, *IEEE Transactions on Information Theory* 14: 515–516.
- Iglezakis, I., Reinartz, T. & Roth-Berghofer, T. (2004). Maintenance memories: Beyond concepts and techniques for case base maintenance, *Proceedings of the 7th European*

- Conference on Case-Based Reasoning*, LNAI 3155, Springer, Madrid, Spain, pp. 227-241.
- Inselberg, A. (1985). The plane with parallel coordinates, *The Visual Computer* 1: 69-91.
- Jarmulak, J., Craw, S. M. & Rowe, R. (2000). Genetic algorithms to optimise CBR retrieval, *Proceedings of the 5th European Workshop on Case-Based Reasoning (EWCBR 2k)*, pp. 136-147.
- Kamada, T. & Kawai, S. (1989). An algorithm for drawing general undirected graphs, *Information Processing Letters* 31(1): 7-15.
- King, R., Feng, C. & Sutherland, A. (1995). Statlog: Comparison of classification algorithms on large real-world problems, *Applied Artificial Intelligence* 9(3): 259-287.
- Kolodner, J. (1983). Reconstructive memory: A computer model, *Cognitive Science* 7(4): 281-328.
- Kolodner, J. (1993). *Case-Based Reasoning*, Morgan Kaufmann, San Mateo, CA.
- Leake, D. B. (1996a). CBR in context: The present and future, in D. B. Leake (ed.), *Case-Based Reasoning: Experiences, Lessons and Future Directions*, AAAI Press/MIT Press, pp. 3-30.
- Leake, D. B. (ed.) (1996b). *Case-Based Reasoning: Experiences, Lessons and Future Directions*, AAAI Press/MIT Press.
- Leake, D. B., Smyth, B., Wilson, D. & Yang, Q. (2001). Maintaining case-based reasoning systems, *Computational Intelligence* 17(2): 193-195.
- Leake, D. B. & Wilson, D. (1998). Categorizing case-base maintenance: Dimensions and directions, *Advances in Case-Based Reasoning, Proceedings of the Fourth European Workshop on Case-Based Reasoning*, Springer-Verlag, pp. 196-207.
- Leake, D. B. & Wilson, D. C. (2000). Remembering why to remember: Performance-guided case-base maintenance, *Proceedings of the 5th European Workshop on Case-Based Reasoning (EWCBR 2k)*, pp. 161-172.

- Liu, B., Ku, L.-P. & Hsu, W. (1997). Discovering interesting holes in data, *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 97)*, pp. 930-935.
- Liu, B., Wang, K., Mun, L.-F. & Qi, X.-Z. (1998). Using decision tree induction for discovering holes in data, *Proceedings of the 5th Pacific Rim International Conference on Artificial Intelligence (PRICAI 98)*, pp. 182-193.
- Lopez de Mantaras et al., R. (2006). Retrieval, reuse, revision and retention in case-based reasoning, *The Knowledge Engineering Review* 20(3): 215-240.
- Maletic, J. & Marcus, A. (2000). Data cleansing: beyond integrity analysis, *In Proceedings of the Conference on Information Quality*, pp. 200-209.
- Markovich, S. & Scott, P. (1988). The role of forgetting in learning, *Machine Learning: Proceedings of the 5th International Conference*, pp. 459-465.
- Massie, S., Craw, S. & Wiratunga, N. (2004a). Visualisation of case-base reasoning for explanation, *Workshop Proceedings of the 7th European Conference on Case-Based Reasoning*, Madrid, pp. 135-144.
- Massie, S., Craw, S. & Wiratunga, N. (2004b). A visualisation tool to explain case-base reasoning solutions for tablet formulation, *In Proceedings of the 24th Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence*, pp. 222-234.
- McArdle, G. P. & Wilson, D. C. (2003). Visualising case-base usage, *Workshop Proceedings of the 5th International Conference on Case-Based Reasoning*, NTNU, Trondheim, Norway, pp. 105-124.
- McCarthy et al. (1955). A proposal for the dartmouth summer research project on artificial intelligence.
- McCarthy, K., Reilly, J., Smyth, B. & McGinty, L. (2005). Generating diverse compound critiques., *Artif. Intell. Rev.* 24(3-4): 339-357.

- McKenna, E. & Smyth, B. (1998). A competence model for case-based reasoning, *In 9th Irish Conference on Artificial Intelligence and Cognitive Science*.
- McKenna, E. & Smyth, B. (2000). Competence-guided case-base editing techniques, *In Proceedings of the 5th European Workshop on Case-Based Reasoning*, pp. 186-197.
- McKenna, E. & Smyth, B. (2001a). Competence-guided case discovery, in M. Dramer, F. Coenen & A. Prece (eds), *Proceedings of the 21st (BCS SGES) International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, Springer-Verlag, Cambridge, UK, pp. 97-108.
- McKenna, E. & Smyth, B. (2001b). An interactive visualisation tool for case-based reasoners, *Applied Intelligence* 14(1): 95-114.
- McSherry, D. (1999). Relaxing the similarity criteria in adaptation knowledge, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99)*, pp. 56-61.
- McSherry, D. (2000). The case-recognition problem in intelligent case-authoring support, *11th Irish Conference on Artificial Intelligence and Cognitive Science*, pp. 180-189.
- McSherry, D. (2001). Intelligent case-authoring support in casemaker-2, *Computational Intelligence* 17(2): 331-345.
- McSherry, D. (2002). Diversity-conscious retrieval, in S.Craw & A.Prece (eds), *Proceedings of the 6th European Conference on Case-Based Reasoning*, LNAI 2416, Springer-Verlag, pp. 219-233.
- McSherry, D. (2004). Explaining the pros and cons of conclusions in CBR, *Proceedings of the 7th European Conference on Case-Based Reasoning*, Madrid, pp. 317-330.
- Minton, S. (1990). Qualitative results concerning the utility of explanation-based learning, *Artificial Intelligence* 42: 363-391.
- Mitchell, T. M. (1997). *Machine Learning*, McGraw-Hill.
- Mullins, M. & Smyth, B. (2001). Visualisation methods in case-based reasoning, *Workshop Proceedings of the 4th International Conference on Case-Based Reasoning*.

- Nugent, C., Cunningham, P. & Doyle, D. (2005). The best way to instil confidence is by being right; an evaluation of the effectiveness of case-based explanations in providing user confidence, *Proceedings of 6th International Conference on Case-Based Reasoning*, pp. 368-381.
- Patterson, D., Anand, S., Dubitzky, D. & Hughes, J. (2000). A knowledge light approach to similarity maintenance for improving case-based competence, *In Proceedings of Workshop on Flexible Strategies for Maintaining Knowledge Containers at 14th European Conference on Artificial Intelligence*, pp. 65-77.
- Patterson, D., Rooney, N. & Galushka, M. (2002). Towards dynamic maintenance of retrieval knowledge in CBR, *In Proceedings of the 15th International FLAIRS Conference*, AAAI Press, pp. 126-131.
- Patterson, D., Rooney, N. & Galushka, M. (2003). Efficient real time maintenance of retrieval knowledge in case-based reasoning, *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 03)*, LNAI 2689, Springer, Trondheim, Norway, pp. 407-421.
- Portinale, L., Torasso, P. & Tavano, P. (1999). Speed-up, quality and competence in multi-modal case-based reasoning, *in L. B. K.D. Althoff, R. Bergmann (ed.), Proceedings of the 3rd International Conference on CBR*, Springer, pp. 303-317.
- Richter, M. (1998). Introduction, *in M. Lenz, B. Bartsch-Sporl & S. Wess (eds), Case-Based Reasoning Technology: From Foundations to Applications*, LNAI 1400, Springer, pp. 1-15.
- Riesbeck, C. & Schank, R. (1989). *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates, Northvale, NJ.
- Ritter, G., Woodruff, H. B., Lowry, S. & Isenhour, T. (1975). An algorithm for a selective nearest neighbor decision rule, *IEEE Transactions on Information Theory* 21(6): 665-669.
- Roth-Berghofer, T. (2004). Explanations and case-based reasoning: foundational issues,

- in P. Funk & P. A. G. Calero (eds), *Proceedings of the 7th European Conference on Case-Based Reasoning*, Springer-Verlag, pp. 389–403.
- Salzberg, S. (1991). A nearest hyperrectangle learning method, *Machine Learning* 6: 227–309.
- Schank, R. (1982). *Dynamic Memory: a Theory of Reminding and Learning in Computers and People*, Cambridge University Press.
- Schank, R. & Abelson (1977). *Scripts, Plans, Goals and Understanding*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Smyth, B. & Cunningham, P. (1996). The utility problem analysed: A case-based reasoning perspective, *Proceedings of the 3rd European Workshop on Case-Based Reasoning (EWCBR 96)*, pp. 392–399.
- Smyth, B. & Keane, M. T. (1995). Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pp. 377–382.
- Smyth, B. & Keane, M. T. (1996). Adaptation-guided retrieval: Using adaptation knowledge to guide the retrieval of adaptable cases, *In Proceedings of the 2nd UK Workshop on CBR*, pp. 2–15.
- Smyth, B. & McKenna, E. (1998). Modelling the competence of case-bases, *Lecture Notes in Computer Science* 1488: 208–220.
- Smyth, B. & McKenna, E. (1999a). Building compact competent case-bases, in K.-D. Althoff, R. Bergmann & L. K. Branting (eds), *Proceedings of the 3rd International Conference on CBR*, LNAI 1650, Springer-Verlag, pp. 329–342.
- Smyth, B. & McKenna, E. (1999b). Footprint-based retrieval, in K.-D. Althoff, R. Bergmann & L. K. Branting (eds), *Proceedings of the 3rd International Conference on CBR*, LNAI 1650, Springer-Verlag, pp. 343–357.
- Smyth, B., Mullins, M. & McKenna, E. (2000). Picture perfect: Visualisation techniques

- for case-based reasoning, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI2000)*, Berlin, Germany, pp. 65-69.
- Sormo, F., Cassens, J. & Aamodt, A. (2005). Explanation in case-based reasoning perspectives and goals, *Artificial Intelligence Review* 24(2): 109-143.
- Tomek, I. (1976a). An experiment with the edited nearest-neighbour rule, *IEEE Transactions on Systems, Man, and Cybernetics* 6(6): 448-452.
- Tomek, I. (1976b). Two modifications of cnn, *IEEE Transactions on Systems, Man, and Cybernetics* 7(2): 679-772.
- Turing, A. (1950). Computing machinery and intelligence, *Mind* 59: 433-460.
- Wettschereck, D. & Dietterich, T. G. (1995). An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms, *Machine Learning* 19(1): 5-27.
- Wilson, D. (1972). Asymptotic properties of nearest neighbour rules using edited data, *IEEE Transactions on Systems, Man, and Cybernetics* 2(3): 408-421.
- Wilson, D. R. & Martinez, T. R. (1997). Instance pruning techniques, *Machine Learning: Proceedings of the 14th International Conference*, Morgan Kaufmann, pp. 403-411.
- Wilson, D. R. & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms, *Machine Learning* 38(3): 257-286.
- Wiratunga, N., Craw, S. M. & Rowe, R. (2002). Learning to adapt for case-based design, *Proceedings of the 6th European Conference on Case-Based Reasoning*, pp. 423-437.
- Wiratunga, N., Craw, S. & Massie, S. (2003). Index driven selective sampling for cbr, *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 03)*, LNAI 2689, Springer, Trondheim, Norway, pp. 637-651.
- Witten, I. H. & Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Diego, CA.
- Woon, Knight & Pedridis (2003). Case base reduction using solution-space metrics, *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 03)*, LNAI 2689, Springer, pp. 652-664.

- Zehraoui, Kanawati & Salotti (2003). Case base maintenance for improving prediction quality, *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 03)*, LNAI 2689, Springer, Trondheim, Norway, pp. 703-717.
- Zhang, J. (1992). Selecting typical instances in instance-based learning, *Machine Learning: Proceedings of the 9th International Conference*, pp. 470-479.
- Zhu, J. & Yang, Q. (1999). Remembering to add: Competence-preserving case-addition policies for case base maintenance, *IJCAI*, pp. 234-241.
- Zhu, X. & Wu, X. (2004). Class noise vs attribute noise: a quantitative study of their impacts, *Artificial Intelligence Review* 22: 177-210.

Appendix A

Published Papers

- S. Massie, S. Craw, and N. Wiratunga. Complexity profiling for informed Case-Base Editing. In *Proceedings of the 8th European Conf. on Case-Based Reasoning*, pages 325–329, Springer, 2006.
- N. Wiratunga, R. Lothian, and S. Massie. Unsupervised Feature Selection for Text Data. In *Proceedings of the 8th European Conf. on Case-Based Reasoning*, pages 340–354, Springer, 2006.
- N. Wiratunga, S. Massie, S. Craw, A. Donati, and E. Vicari. Case Based Reasoning for Anomaly Report Processing. In *Proceedings of 3rd Textual Case-Based Reasoning Workshop at the 8th European Conf. on Case-Based Reasoning*, pages 44–49, 2006.
- S. Massie, S. Craw, and N. Wiratunga. Complexity guided case discovery for case based reasoning. In *Proceedings of the 20th National Conf. on Artificial Intelligence*, pages 216–221, AAAI Press, 2005.
- S. Massie, S. Craw, and N. Wiratunga. A Visualisation Tool to Explain Case-Base Reasoning Solutions for Tablet Formulation. In *Proceedings of the Twenty Fourth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 222–236, Springer, 2004.
- N. Wiratunga, I. Koychev, and S. Massie. Feature Selection and Generalisation for Retrieval of Textual Cases. In *Proceedings of the 7th European Conference on Case-based reasoning*, pages 806–820, Springer, 2004. Best Paper Award.

- S. Massie, S. Craw, and N. Wiratunga. Visualisation of Case-Base Reasoning for Explanation. In *Workshop Proceedings of the 7th European Conference on Case-based reasoning*, pages 135–144, 2004.
- S. Massie, S. Craw, and N. Wiratunga. What is CBR Competence? In *Poster Presentation at the twenty-third Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence* (published in BCS Expert Update 8(1):7-10) 2003.
- N. Wiratunga, S. Craw, and S. Massie. Index Driven Selective Sampling for Case-Based Reasoning. In *Proceedings of the Fifth International Conference on Case-Based Reasoning*, pages 637–651, Springer, 2003.

Complexity Profiling for Informed Case-Base Editing

Stewart Massie, Susan Craw, and Nirmalie Wiratunga

School of Computing,
The Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{sm, smc, nw}@comp.rgu.ac.uk

Abstract. The contents of the case knowledge container is critical to the performance of case-based classification systems. However the knowledge engineer is given little support in the selection of suitable techniques to maintain and monitor the case-base. In this paper we present a novel technique that provides an insight into the structure of a case-base by means of a complexity profile that can assist maintenance decision-making and provide a benchmark to assess future changes to the case-base. We also introduce a complexity-guided redundancy reduction algorithm which uses a local complexity measure to actively retain cases close to boundaries. The algorithm offers control over the balance between maintaining competence and reducing case-base size. The ability of the algorithm to maintain accuracy in a compacted case-base is demonstrated on seven public domain classification datasets.

1 Introduction

Case-Based Reasoning (CBR) is an experience based problem-solving approach that uses a case-base of previously solved problems as a knowledge source to help solve new problems. The case-base is a key knowledge container [13] and, as such, the CBR process draws heavily on case knowledge. This is particularly true in case-based classification systems for which retrieval is the key stage.

The CBR paradigm typically employs a lazy learning approach, such as k -nearest neighbour [5], for the retrieval stage of the process which delays generalisation until problem-solving time. This is attractive because training is not necessary, learning is fast and incremental, algorithms are simple and intuitive, and advance knowledge of the problems to be faced is not required. However, with large case-bases, the drawbacks of lazy learning are high memory requirements since all examples are stored, slow retrieval times, and the possible inclusion of harmful cases.

At the initial case authoring stage, the case-base can consist of all available examples. Alternatively, the knowledge engineer can create a hand-crafted case-base by storing only selected examples in the case-base giving rise to a need for algorithms that control the size of the case-base. In addition, the case-base gets larger over time, often as a result of indiscriminate storage of cases during the retain stage of the CBR cycle. The cases may be redundant and provide no improvement in competence or may even be harmful, noisy cases that result in a reduction in competence. In either case the inclusion of additional cases will increase storage requirements and retrieval times. The cost of retrieval can grow to the extent that it outweighs the benefit of additional cases. This

is called the *utility problem* [7,14] and results in an ongoing requirement to control case-base growth.

Understandably, there has been considerable research on the case-base editing problem giving the knowledge engineer a choice of potential approaches. However, most contemporary editing algorithms give no control over the size of the edited case-base or the impact on competence, and provide no explanation of their decisions. We argue that the knowledge engineer should have more control over the balance between the reduction in the size of the case-base and maintaining competence.

It is often assumed that any of the numerous maintenance approaches available will work well on all domains. However, research has identified that no one algorithm is *best* in all situations [4,6]. The knowledge engineer must make a choice between alternative techniques based on knowledge of the case-base and the system's competence, retrieval and storage space requirements. What technique to choose is not obvious because it requires knowledge about the structure of the case-base that is often hidden. Given a dataset it is not clear what level of redundancy or noise it contains. Low accuracy may be the result of a lack of case knowledge due to a sparse case-base, a difficult problem with long, complex decision boundaries or noisy data. The knowledge engineer does not know whether to apply a noise reduction algorithm, a case creation algorithm or a redundancy reduction algorithm. Methods that improve the comprehension of the case-base structure would aid this decision-making process.

In this paper, we present a novel case-base profiling technique that provides an insight into the structure of the case-base, assisting informed maintenance decisions. In addition, we introduce a new case-base editing algorithm that gives the knowledge engineer more control of the balance between case-base size and competence and also provides some explanation of its editing decisions. Both techniques are evaluated experimentally and shown to have benefit.

The remainder of this paper describes our approach and evaluates it on several public domain case-bases. In Section 2 we review existing research on case-based editing techniques. Section 3 discusses complexity profiling of a case-base and how it can aid the knowledge engineer make maintenance choices. An evaluation of our profiling technique is presented in Section 4. Our new case-base editing technique is then introduced in Section 5 with experimental results being reported on seven datasets in Section 6. Finally, we provide conclusions and recommendation for future work in Section 7.

2 Related Work on Case-Base Editing in CBR

Considerable research effort has been aimed at case-base maintenance and much of the research has focused on control of the case-base by case deletion or case selection policies. Two distinct areas have been investigated: the control of noise; and the reduction of redundancy.

Noise reduction algorithms aim to improve competence by removing cases that are thought to have a detrimental effect on accuracy. These may be corrupt cases with incorrect solutions or, alternately, they may be cases whose inclusion in the case-base results in other cases being incorrectly solved. These algorithms usually remove only a few cases. Wilson Editing [20], also called ENN, is the best known algorithm and

attempts to remove noise by removing cases that are incorrectly classified by their nearest neighbours. ENN removes noisy cases but also deletes cases lying on boundaries between classes leaving smoother decision boundaries. Tomek extends ENN with the Repeated Wilson Editing method (RENN) and the All k -NN method [18]. RENN extends ENN by repeating the deletion cycle until no more cases are removed. The All k -NN is similar, except that after each iteration the value of k is increased. The Blame-Based Noise Reduction (BBNR) algorithm [6] is a noise reduction algorithm that takes a slightly different approach in attempting to identify cases that cause misclassification and removing them if they cause more *harm* than *good*. Noise reduction can reduce competence, hence careful consideration should be given to the domain and structure of the case-base before applying these algorithms to ensure there is a need for noise reduction. Our work does not advance research on noise reduction but rather identifies datasets where noise reduction is required.

Redundancy reduction algorithms can be either incremental, starting with an empty edited set and selecting cases, or decremental where cases are deleted from an initially complete set. Hart's [8] Condensed Nearest Neighbour rule (CNN) was an early incremental approach in which only cases not solved by the edited set are added to it. CNN is sensitive to the case presentation order and numerous extensions or modifications have suggested improvements [1,19]. McKenna and Smyth's [11,17] competence-guided editing techniques use local case information from their competence model [16] to rank cases prior to case selection, so that redundant cases are presented later in the editing process. Several ranking measures are proposed based on a case's *coverage* and *reachability* sets including *relative cover* ranking (RC), which is shown to give a large reduction in case-base size while retaining competence. McKenna and Smyth also developed the CASCADE authoring system [12] in which the case-base developer, guided by a model of case competence, could interact with an interface to manage the selection of which cases to add or remove from the case-base.

Several contemporary decremental approaches use similar local case competence knowledge to guide their editing decisions. Wilson and Martinez's Reduction Technique range of algorithms (RT1-3) [21] is guided by a case's *associates*. The associates of a case is the set of cases which have that case as one of their nearest neighbours and is analogous to Smyth & Keane's [15] coverage set. The algorithms remove a case if at least as many of its associates would be correctly classified after deletion. Brighton and Mellish [3] adopt a similar approach with their Iterative Case Filtering algorithm (ICF). A case is deleted if its reachable set is larger than its coverage set, i.e., more cases can solve the case than it can solve itself. The process is repeated until no more cases are removed. This results in boundary cases being retained and central cases being removed. Delaney and Cunningham [6] employ a similar approach in their Conservative Redundancy Reduction algorithm (CRR) in which a case with smallest coverage set is selected first and any cases that it solves are deleted from the training set. This algorithm was tested on email classification where it is shown that conservative redundancy reduction achieves a higher accuracy than comparable but more aggressive algorithms.

Redundancy reduction algorithms require a trade off between the level of compaction and competence preservation. The more modern algorithms (RC, RT3, ICF and CRR) all provide a good but different balance between these conflicting objectives. Our approach

gives the knowledge engineer control of this balance. The contemporary redundancy editing approaches all rely on models [10,15,16] of the case-base to supply local information about the relationship between cases. These relationships are used to *indirectly* retain cases on decision boundaries. In our approach we also calculate local case information but the information identifies the position of a case in relation to a decision boundary. We aim to *directly* identify and retain cases on or near decision boundaries.

3 Case-Base Complexity Profiling

Our objective is to help the knowledge engineer make decisions on maintenance strategies by providing a global case-base measure of accuracy, noise and redundancy plus local information on the structure of the case-base. Our approach is to provide a profile of a local case metric. We use a case complexity measure to provide the local measure and a ranked profile of this measure to provide a view of the overall effect within the case-base. The complexity profile identifies the mix of local complexities. In the rest of this section we first define the local case complexity measure used and then look at our profiling approach to providing a global picture of the case-base.

3.1 Complexity Measure

The foundation of our approach is to measure the local complexity based on the spatial distribution of cases within the case-base. Complexity is calculated using a metric based on the composition of its neighbours while incrementally increasing the size of its neighbourhood.

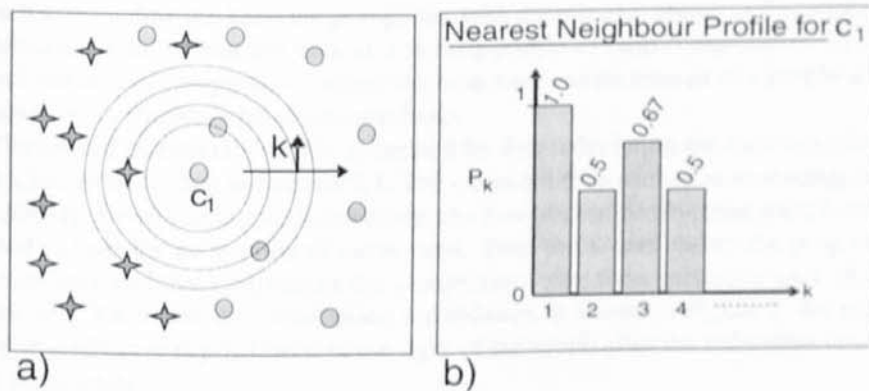


Fig. 1. Calculation of the complexity metric

The complexity measure is calculated for each case by looking at the class distribution of cases within its local neighbourhood. P_k is the proportion of cases within a case's k nearest neighbours that belong to the same class as itself. In Figure 1(a) a case is represented by a symbol on the plot with the class of the case distinguished by the shape, star or circle. If we consider case c_1 , then as the value of k increases, the sequence of P_k starts 1, 0.5, 0.67, 0.5. A nearest neighbour profile can now be plotted

for c_1 using P_k as k increases. The complexity metric is based on the area of the graph under the profile, the shaded area in Figure 1(b). Case complexity is calculated by

$$\text{complexity} = 1 - \frac{1}{K} \sum_{k=1}^K P_k$$

for some chosen K . With $K=4$ the complexity of c_1 is 0.33. A large value for K has little impact on the results because the metric is biased towards a case's nearest neighbours. We have used $K=10$ in our calculations for all but the smallest case-base sizes.

Cases with high complexity are close to classification boundaries and identify areas of uncertainty within the problem space. Cases with complexity greater than 0.5 are closer to cases of a different class than those of their own class, and are potentially noisy. Cases with low complexity are surrounded mainly by cases with the same class as themselves, and are located in areas of the problem space in which the system would be more confident in making a decision on the class of a new problem. Cases with a zero complexity value are surrounded by a sizeable group of cases with the same class as itself, and may be considered redundant because other cases in the group would be able to solve new problems in this region of the problem space.

3.2 Profile Approach

The complexity measure provides a local indicator of uncertainty within the problem space and has been shown to be useful in informing a case discovery algorithm [9]. However, it is difficult for the knowledge engineer to use this local information directly to gain an insight into the structure of a case-base from a global perspective. Our approach to providing the knowledge engineer with meaningful access to this pool of local information is to present the data as a ranked profile of case complexities. In this approach the mix of complexities within the case-base can be viewed as a profile allowing comparisons to be made between case-bases.

The ranked complexity profile is created by first calculating the case complexity of each case, as described in Section 3.1. The cases are then ranked in ascending order of complexity. Then, starting with cases with the lowest complexity, case complexities are plotted against the proportion of cases used. Thus the x-axis shows the proportion of the case-base and the y-axis gives the complexity value for a particular case. A typical profile plot, for a case-base containing redundancy, is shown in Figure 2. An exponentially shaped curve is positioned to the right of the graph after the redundant cases with zero complexity.

Three key global indicators can be taken from this plot to give a measure of accuracy, redundancy and noise respectively, as follows:-

- **Error Rate:** The area under the curve, shown as the shaded area on the plot, gives the overall complexity of the problem being faced and provides a measure of expected error rate.
- **Redundancy:** The position at which the plot breaks away from the x-axis, shown on the profile as x_1 , gives a measure of the level of redundancy within the case-base. This is a measure of the proportion of cases located in single class clusters.

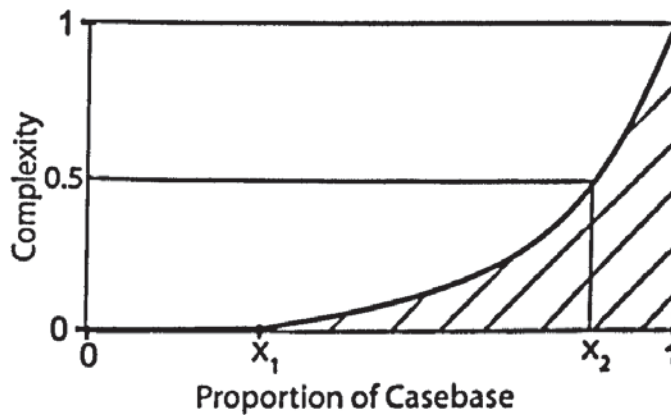


Fig. 2. Typical graph of local complexity profile

- **Noise:** A case with a complexity greater than 0.5 has the majority of its neighbours belonging to a different class. These cases can be considered noisy. The proportion of noisy cases can now be portrayed as the distance from x_2 to 1; i.e., $1-x_2$.

It is expected that these three indicators will correlate well with typical measures of error rate, redundancy and noise. This will be investigated in the evaluation that follows. However, it is the graph itself that provides the best insight into the structure of the case-base, and allows informed decisions to be made by the knowledge engineer in relation to whether the number of cases in the case-base is appropriate to the domain and its level of complexity.

4 Experimental Evaluation of Profiling

We evaluate complexity profiles on two levels in this section. First we examine whether complexity profiling can provide useful comparisons of case-bases from different domains. Then we investigate our hypothesis that the complexity profile indicators accurately predict global error rates and levels of noise and redundancy.

4.1 Cross Domain Comparisons

In the previous section we looked at a *typical* complexity profile and claimed that this profiling provided a good approach at making comparisons across different domains. To examine this claim we look at example complexity profiles from four domains. Figure 3(a)-(c) show the complexity profiles for three public domain classification datasets from the UCI ML repository [2], together with the complexity profile for an artificial dataset in Figure 3(d).

Wine (Figure 3(a)) is a simple three class problem with 14 numeric attributes and 178 instances. It can be seen from the profile that a high level of classification accuracy is expected due to the small area under the complexity curve (0.05). The expected level of noise is very low with an estimate of 4% and a maximum complexity value for an instance being well below 1. A high level of redundancy is also evident with 75% of the

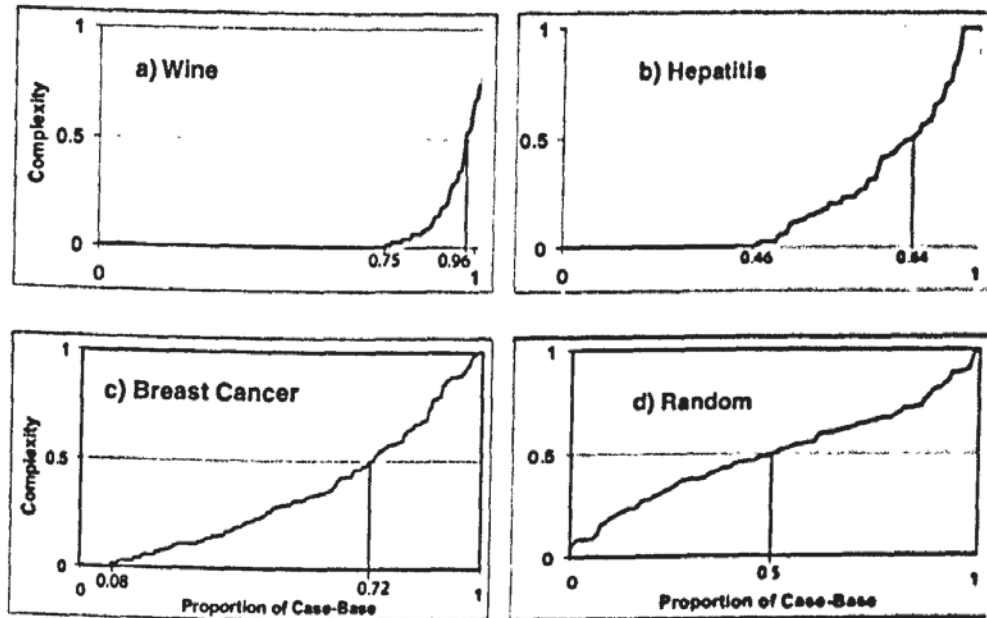


Fig.3. Complexity profiles for sample datasets

instances having a zero complexity value. A case-base created from this dataset, containing less redundancy, could form part of an excellent CBR problem-solver because the similarity measure forms the instances into clusters with the same solutions - similar problems have similar solutions.

Hepatitis (Figure 3(b)) is a smaller dataset of 155 instances represented by 20, mostly nominal, features containing some missing values. This is a more complex problem with an overall complexity of 20% and a gentler slope to the curve than for Wine, suggesting more complex decision boundaries. There is a moderate predicted level of noise (16%) with several instances completely surrounded by instances of an opposing class resulting in a peak complexity value of 1. Although there is less redundancy than for wine, the level is still high with 46% of the instances surrounded by at least 10 instances with the same class. Applying noise reduction algorithms would probably improve the level of accuracy achieved and redundancy reduction algorithms could be applied to reduce storage requirements without affecting accuracy levels.

Breast Cancer (Figure 3(c)) is a binary classification domain with 9 multi-valued features containing missing data. This is a complex problem, with the low slope on the graph indicating most instances lie close to decision boundaries. There is a high estimated level of noise (28%) and little redundancy (8%). This profile would suggest a dataset that is not suitable for a CBR application as it stands. Applying noise reduction algorithms may improve accuracy levels. In addition, improvements in the similarity measure or case representation could be investigated to create a design in which problems with similar solutions are better recognised as being similar.

The final profile, Figure 3(d), is for an artificial dataset with 100 instances. This is a binary classification problem with 2 numerical features where the class of an instance

is randomly selected. This is a problem that has been created so that similar problems will not form into a cluster of instances with similar solutions. The dataset would not make a suitable case-base for a case-based problem-solver and this is confirmed by the complexity profile. As expected, the predicted error rate is 50% and the predicted noise level is also 50% because instances are as likely to be surrounded by instances of an opposing class as the same class. There is no redundancy because the instances do not form into large same class clusters.

4.2 Accuracy and Noise Predictions

The evaluation of complexity profiles from different domains, and the insight the profile provides, assumes that the error rate, noise level and redundancy level indicators are good predictors of the real values contained within the data. While conceptually the use of these indicators appears reasonable, we want to investigate the relationships empirically.

Table 1. Results summary of complexity profile indicators compared to alternative measures

Case-Base	ERROR RATE		NOISE		REDUNDANCY
	TEST SET	PROFILE	ENN	PROFILE	PROFILE
Wine	0.037	0.050	0.033	0.04	0.75
Iris	0.059	0.058	0.048	0.05	0.79
Hepatitis	0.189	0.203	0.176	0.16	0.46
Lymphography	0.187	0.242	0.155	0.14	0.23
Breast Cancer	0.339	0.344	0.306	0.28	0.08
House Votes	0.079	0.083	0.071	0.07	0.77
Zoo	0.038	0.085	0.061	0.06	0.70

Accuracy or error rate is the easiest indicator to compare. We calculate error rate experimentally using ten fold cross-validation. Nine folds are retained as the training set with the remaining fold being the unseen test set. The average error rates for seven UCI datasets, calculated using 1-NN, are shown in column 2 of Table 1 with the corresponding error rate indicator from the complexity profiles shown in column 3. There is a strong correlation between the results as can be seen by the close fit to the straight line in Figure 4, which plots the complexity profile prediction against test set error rate.

There is not an obvious measure of noise with which to make a comparison. However, ENN is the best known noise reduction algorithm. Hence, we use the reduction in the size of a dataset after applying ENN as a benchmark measure of noise with which to compare our predicted indicator from the complexity profile. The average edited set size after applying ENN as a proportion of the original dataset size is shown in column 4 of Table 1. This is compared with the average complexity profile noise indicator, as shown in column 5. Again there is a strong correlation between the results, as shown by the fit to a straight line in Figure 5, which plots the complexity profile noise prediction with the proportional reduction in the size of the dataset from applying ENN.

These results confirm that the complexity profile is a good predictor of accuracy and noise. The ability of the complexity profile to predict redundancy is difficult to measure directly but is investigated in more detail in Section 5.

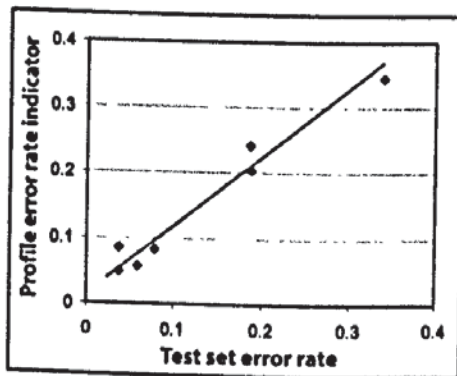


Fig. 4. Error rate correlation

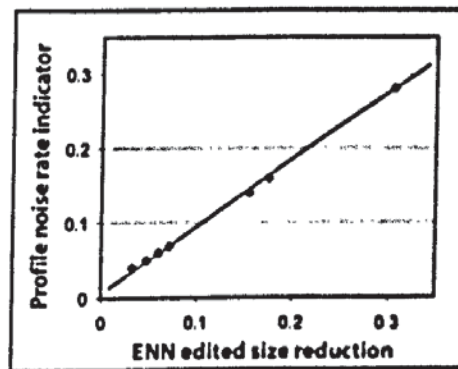


Fig. 5. Noise level correlation

5 Complexity Threshold Editing

The case-base complexity profile provides a tool that can be used for informed redundancy editing in which the knowledge engineer has control over the level of redundancy reduction. As with most redundancy editing algorithms, our approach aims to give a high classification accuracy and to provide significant storage space reduction. However, these objectives can be contradictory. Aggressive case editing can achieve large reductions in case-base size but at the expense of classification accuracy [6]. The complexity profile provides a measure of the proportion of redundant cases compared to cases near decision boundaries giving an explanation of the effect of different levels of redundancy reduction on competence.

In classification problems redundant cases are found in clusters with the same classification preferably far from decision boundaries. Our approach to case-base editing is to identify and delete redundant cases while at the same time retaining boundary cases. The complexity measure, described in Section 3.1, is a good identifier of boundary cases, with a high complexity value, and redundant cases with a low complexity. We use the local case complexity to guide our editing algorithm.

The benefits of this approach over existing techniques are two-fold. Firstly, the knowledge engineer is in control of the maintenance process and is able to make an informed decision on a suitable level of case-base compaction dependent on a system's performance requirements. This decision is not made by selecting an arbitrary case-base size. Rather, through a review of the complexity profile, a judgement can be made on the impact of different complexity thresholds. If storage space or retrieval time requirements are crucial to the design a higher threshold can be chosen in the understanding that it will reduce competence. Secondly, the complexity profile provides an explanation of the editing process by providing a transparency to the process and a justification for deleting the selected cases.

The basic approach is to set a complexity threshold and delete cases with a complexity below the threshold. The threshold is set on the y-axis, and the resulting size of the case-base can be noted on the x-axis. Our expectation is that setting a zero threshold will remove only cases that are likely to be redundant and not result in a fall in competence. Competence is expected to decline gradually as the complexity threshold is increased. The basic approach gave promising results but also highlighted several problems.

- Noisy cases are by their very nature boundary cases and hence will be retained by this algorithm. Adopting the approach of most other contemporary editing algorithms, we add a pre-processing noise editing algorithm (RENN).
- Clusters of cases all with zero complexity can form. Setting a simple threshold can delete the complete cluster. It would be better to retain at least one case to represent the cluster. To overcome this problem an iterative approach is employed with case complexities being recalculated after each case deletion and the cases being re-ranked in ascending order of complexity.
- A further problem with clusters of cases with zero complexity is the choice of order of deletion. If a random selection is made cases nearer decision boundaries may be selected for deletion first. This would harm the performance of the algorithm so we introduce a *friend* to *enemy* distance ratio as a secondary ranking. The friend distance is the average distance to the case's nearest like neighbours whereas the enemy distance is the average distance to the case's nearest unlike neighbours. A high ratio indicates a case closer to a decision boundary and farther from cases of the same class, whereas a low ratio indicates a case farther from a decision boundary and in a cluster of cases of same class.

The Complexity Threshold Editing algorithm (CTE), incorporating the changes introduced above, is described in Figure 6.

```

T,      Dataset of n cases (c1 ... cn)
COM(S), Calculate case complexity, distance
        ratio and order cases in set S
RENN(S), Apply noise removal to set S
Count=0

COM(T)
For each c in T
  If ( complexity(c)<threshold) count++
End-For
E-Set ← RENN(T)
For 0 to count
  COM(E-Set)
  c ← First case in E-Set
  E-Set ← E-Set - c
End-For
Return (E-Set)

```

Fig. 6. Complexity threshold editing algorithm

6 Experimental Evaluation of Complexity Threshold Editing

In this evaluation we compare the performance of Complexity Threshold Editing with several existing redundancy reduction algorithms. The algorithms being compared can be split into three categories.

- Complexity Threshold Editing (CTE) is our new redundancy reduction algorithm and directly comparable with existing redundancy reduction algorithms. It is evaluated with four different complexity thresholds (0, 0.1, 0.2 and 0.3)
- Existing redundancy reduction algorithms (CRR, ICF and RC). These are modern redundancy reduction algorithms that aim to reduce the size of the case-base while maintaining competence. They have been shown to perform well in previous comparisons but each provides a different balance between compaction and competence: CRR provides a conservative approach to redundancy editing, RC is an aggressive algorithm deleting the highest number of cases, whereas, ICF falls in the middle giving a moderate level of case-base compaction.
- Noise reduction algorithms (ENN and RENN). These algorithms aim to improve competence but remove only a few cases and are not comparable with redundancy reduction algorithms. They are included in the evaluation because RENN has been used as the pre-processing algorithm for all the redundancy reduction algorithms including CTE. RENN provides a benchmark for accuracy that the redundancy reduction algorithms aim to maintain.

6.1 Experimental Setup

A ten times 10-fold cross validation experimental set-up is used giving one hundred case-base/test set combinations per experiment. The editing algorithms were applied to each case-base and the resulting edited set size is recorded. Test set accuracy, using 1-NN retrieval, was measured for the original case-base and for each of the edited sets created by the editing algorithms.

Comparisons have been made on seven UCI datasets and the averaged results are shown in Tables 2 and 3. Table 2 contains the average test set accuracy for each algorithm on each domain. The highest accuracy result achieved by the redundancy reduction algorithms in each domain is highlighted in bold. Table 3 gives the unedited dataset size in column 2 together with the edited dataset size as a proportion of the original in the other columns. The values in bold are the size reduction achieved by the redundancy algorithm with the highest accuracy. Both tables include an *average* row but this should be used with care as it is calculated across different domains.

6.2 Results of Evaluation

The results of the evaluation can be summarised in each of the categories as follows:

- The CTE algorithm provides the highest accuracy of the redundancy reduction algorithms in six of the seven domains. At zero complexity threshold, CTE has the highest average accuracy of 87.5% compared to 86.5% for CRR. This is achieved with smaller case-base sizes, 32.8% of original size on average compared to 39.4%,

Table 2. Comparison of average test set accuracy for alternative editing algorithms

Case-Base	ORIG	REDUNDANCY			CTE				NOISE	
		CRR	ICF	RC	0	0.1	0.2	0.3	RENN	ENN
Breast Cancer	0.661	0.740	0.736	0.688	0.738	0.734	0.728	0.709	0.753	0.736
Hepatitis	0.808	0.839	0.833	0.821	0.859	0.853	0.847	0.822	0.834	0.862
House Votes	0.922	0.905	0.901	0.904	0.922	0.916	0.898	0.854	0.911	0.920
Iris	0.940	0.947	0.931	0.943	0.949	0.933	0.882	0.878	0.952	0.952
Lymphography	0.812	0.759	0.749	0.757	0.775	0.775	0.776	0.758	0.772	0.781
Wine	0.963	0.957	0.934	0.923	0.959	0.924	0.884	0.822	0.948	0.953
Zoo	0.957	0.906	0.902	0.904	0.921	0.901	0.876	0.778	0.904	0.926
Average	0.866	0.865	0.855	0.850	0.875	0.862	0.842	0.803	0.868	0.876

Table 3. Comparison of edited case-base size for alternative editing algorithms

Case-Base	ORIG	REDUNDANCY			CTE				NOISE	
		CRR	ICF	RC	0	0.1	0.2	0.3	RENN	ENN
Breast Cancer	258	0.248	0.160	0.071	0.604	0.450	0.292	0.163	0.674	0.694
Hepatitis	140	0.403	0.082	0.061	0.355	0.265	0.186	0.102	0.796	0.824
House Votes	392	0.471	0.035	0.038	0.155	0.093	0.061	0.038	0.908	0.928
Iris	135	0.389	0.296	0.065	0.177	0.078	0.038	0.037	0.952	0.952
Lymphography	134	0.415	0.180	0.152	0.625	0.460	0.322	0.189	0.815	0.846
Wine	161	0.439	0.159	0.099	0.208	0.098	0.056	0.033	0.965	0.967
Zoo	91	0.355	0.486	0.110	0.241	0.138	0.096	0.075	0.927	0.938
Average	187	0.394	0.153	0.074	0.328	0.221	0.146	0.087	0.853	0.870

showing that CTE is an excellent algorithm for conservative redundancy reduction. At moderate levels of redundancy reduction, with a threshold of 0.1, CTE achieves slightly better accuracies than ICF but retains slightly more cases. Overall the performance is comparable with ICF. With higher complexity thresholds, for aggressive redundancy reduction, CTE does not perform so well and is outperformed by RC.

- The three existing redundancy reduction algorithms all provide a different compromise on the trade-off between case-base compaction and maintaining competence. CRR, designed to take a conservative approach to redundancy reduction, has the highest accuracy on each domain and the highest average accuracy of 86.5% compared to 85.5% for ICF and 85.0% for RC. However, CRR obtains the improved accuracy by retaining, on average, 39% of the cases, more than twice that of ICF (15%) and five times RC (7%). Very aggressive redundancy reduction is achieved by RC but the results confirm that this is at the expense of loss of accuracy. The

performance of ICF lies between the others on both competence retention and case-base size reduction.

- There is little to choose between the performance of the noise reduction algorithms. In these datasets ENN gives the highest average accuracy but that is probably because many of these datasets are not noisy and ENN gives the best results on data with low levels of noise. RENN removes slightly more cases and generally performs better on noisy data but worse on low noise datasets. It is worth noting that on four of the datasets all the noise reduction algorithms harm accuracy but on Breast Cancer and Hepatitis substantial accuracy gains are achieved by noise reduction.

CTE provides the best performance for conservative redundancy reduction, providing superior accuracy on six out of the seven domains. We checked the significance of these differences using a 2-tailed t-test with 95% confidence level. The superiority of CTE was found to be significant in 4 domains; Hepatitis, House Votes, Lymphography and Zoo.

As expected, setting a zero level threshold maintained accuracy at a similar level to that achieved after RENN noise reduction in all the domains and overall there was actually a slight increase in accuracy from 86.8% to 87.5%. This confirms that at the local level the case complexity measure identifies redundant cases and at a global level the redundancy indicator estimated from the complexity profile is a good predictor of the level of redundancy within a case-base. When the complexity threshold is increased above zero, accuracy initially falls away gradually at first, as non-redundant cases start to be deleted and then more quickly as cases nearer to decision boundaries are deleted.

The performance of CTE for aggressive levels of redundancy reduction with the higher complexity thresholds was disappointing. This suggests that while case complexity provides a good measure for identifying redundant cases away from boundaries, it is not so good at selecting between boundary cases.

The expectation that accuracy would fall as the size of the edited case-base falls is corroborated both for the existing redundancy reduction algorithms and for varying complexity thresholds with CTE. This confirms previous research results that there is a trade-off between the conflicting objectives of compaction of the case-base and maintaining competence.

The inconsistent performance of the noise removal algorithms across the different datasets highlights the need to apply different maintenance strategies for different domains. Complexity profiling of the case-base can play a role in identifying appropriate maintenance strategies for a case-base.

7 Conclusions and Future Work

The novel contribution of this work is the use of a local case complexity measure, together with a case-base profile to guide the case editing process. The complexity measure identifies redundant cases for deletion and cases on class decision boundaries for retention. Complexity profiling gives a measure of the level of complexity, redundancy and noise inherent in the data. This knowledge provides an element of control over the compromise required between the contradictory objectives of the reduction in case-base size and the retention of competence.

Complexity profiling can play a further role in assisting the knowledge engineer to make choices between alternative maintenance techniques depending on the structure of the data and a system's performance requirements. Profiling also provides the opportunity to create a benchmark for comparison with future versions of the case-base to monitor the impact of changes over time.

We have introduced the Complexity Threshold Editing algorithm and demonstrated its effectiveness on seven public domain datasets. The algorithm was shown to provide superior performance characteristics when compared to existing techniques for conservative levels of editing and comparable performance at moderate levels of editing. One limitation of the approach is an average performance for aggressive editing because the complexity measure does not make a balanced selection between alternative boundary cases. Enhancements are being investigated to improve this selection on boundaries.

Complexity profiling has also been introduced and evaluated on public domain datasets. The interpretation of several profiles has been discussed to show how they can help the knowledge engineer develop a suitable case-base maintenance policy. The global indicators on accuracy, redundancy and noise, extracted from the profiles, are shown to correlate well with alternative measures.

In this paper we have concentrated on providing support for the knowledge engineer in the redundancy editing problem. However we are keen to see how the use of profiling might be used more generally to provide support in other case-base maintenance areas, such as noise reduction.

References

1. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. *Machine Learning* 6(1) (1991) 37–66
2. Blake, C., Keogh, E., Merz, C.: UCI repository of machine learning databases. (1998)
3. Brighton, H., Mellish, C.: Identifying competence-critical instances for instance-based learners. In *Instance Selection and Construction for Data Mining* (2001) 77–94
4. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery* 6(2) (2002) 153–172
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1) (1967) 21–27
6. Delany, S.J., Cunningham, P.: An analysis of case-base editing in a spam filtering system. In *Proceedings of the 7th European Conference on Case-Based Reasoning* (2004) 128–141
7. Francis, A., Ram, A.: Computational models of the utility problem and their application to a utility analysis of case-based reasoning. In *Proceedings of the Workshop on Knowledge Compilation and Speed-Up Learning* (1993)
8. Hart, P.: The condensed nearest neighbour rule. *IEEE Transactions on Information Theory*, 14 (1968) 515–516
9. Massie, S., Craw, S., Wiratunga, N.: Complexity-guided case discovery for case based reasoning. In *Proceedings of the 20th National Conference on Artificial Intelligence* (2005) 216–221
10. McKenna, E., Smyth, B.: A competence model for case-based reasoning. In *9th Irish Conference on Artificial Intelligence and Cognitive Science* (1998)
11. McKenna, E., Smyth, B.: Competence-guided case-base editing techniques. In *Proceedings of the 5th European Workshop on Case-Based Reasoning* (2000) 186–197

12. McKenna, E., Smyth, B.: An interactive visualisation tool for case-based reasoners. *Applied Intelligence*, 14(1) (2001) 95–114
13. Richter, M.: Introduction. In *Case-Based Reasoning Technology: From Foundations to Applications* (1998) 1–15
14. Smyth, B., Cunningham, P.: The utility problem analysed: A case-based reasoning perspective. In *Proceedings of the 3rd European Workshop on Case-Based Reasoning* (1996) 392–399
15. Smyth, B., Keane, M.T.: Remembering to forget. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (1995) 377–382
16. Smyth, B., McKenna, E.: Modelling the competence of case-bases. In *Proceedings of the 4th European Workshop on Case-Based Reasoning* (1998) 208–220
17. Smyth, B., McKenna, E.: Building compact competent case-bases. In *Proceedings of the 3rd International Conference on Case-Based Reasoning* (1999) 329–342
18. Tomek, I.: An experiment with the edited nearest-neighbour rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6) (1976) 448–452
19. Tomek, I.: Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(2) (1976) 679–772
20. Wilson, D.: Asymptotic properties of nearest neighbour rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3) (1972) 408–421
21. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3) (2000) 257–286

Unsupervised Feature Selection for Text Data

Nirmalie Wiratunga, Rob Lothian, and Stewart Massie

School of Computing,
The Robert Gordon University,
Aberdeen AB25 11G, Scotland, UK
{nw, rml, sm}@comp.rgu.ac.uk

Abstract. Feature selection for unsupervised tasks is particularly challenging, especially when dealing with text data. The increase in online documents and email communication creates a need for tools that can operate without the supervision of the user. In this paper we look at novel feature selection techniques that address this need. A distributional similarity measure from information theory is applied to measure feature utility. This utility informs the search for both representative and diverse features in two complementary ways: **CLUSTER** divides the entire feature space, before then selecting one feature to represent each cluster; and **GREEDY** increments the feature subset size by a greedily selected feature. In particular we found that **GREEDY**'s local search is suited to learning smaller feature subset sizes while **CLUSTER** is able to improve the global quality of larger feature sets. Experiments with four email data sets show significant improvement in retrieval accuracy with nearest neighbour based search methods compared to an existing frequency-based method. Importantly both **GREEDY** and **CLUSTER** make significant progress towards the upper bound performance set by a standard supervised feature selection method.

1 Introduction

The volume of text content on the Internet and the widespread use of email-based communication have created a need for text classification, clustering and retrieval tools. There is also growing research interest in email applications, both within the Case-Based Reasoning (CBR) community [6,12] and more generally in Machine Learning [15]. Fundamental to this interest is the challenge posed by unstructured content, large vocabularies and changing concepts. Understandably, much of the research effort is directed towards mapping text into structured case representations, so as to facilitate meaningful abstraction, comparison, retrieval and reuse.

Feature selection plays an important role for the indexing vocabulary acquisition task. Often this initial selection can be either directly or indirectly applied to identify representative dimensions with which structured cases can be formed from unstructured text data. Applied directly, each selected feature corresponds to a dimension in the case representation. When applied indirectly, selected features are first combined to identify new features in a process referred to as feature extraction before they can be used as dimensions for case representation [4,25]. Although feature extraction is undoubtedly more effective than feature selection at capturing context, our experiences with supervised tasks suggests that feature selection is an important complementary precursor to

the extraction phase [24]. In this paper we are interested in feature selection applied directly to derive case representations for unsupervised tasks involving text data.

Feature selection reduces dimensionality by removing non-discriminatory and sometimes detrimental features, and has been successful in improving accuracy, efficiency and comprehension of learned models for supervised tasks in both structured [8,10] and unstructured domains [26]. Feature selection in an unsupervised setting is far more challenging, especially when dealing with text data. Typical applications (e.g. email, helpdesk, online reports) involve clustering of text for retrieval and maintenance purposes. The exponential increase in on-line text content creates a need for tools that can operate without the supervision of the user. However, in spite of this need, current research in feature selection is mainly concerned with supervised tasks only.

The aim of this paper is to apply unsupervised feature selection to text data. We introduce feature selection methods that are applicable to free text content as in emails and to texts that are sub-parts of semi-structured problem descriptions. The latter form is typical of reports such as anomaly detection or medical reports. Analysis of similar words and their neighbourhoods provide insight into vocabulary usage in the text collection. This knowledge is then exploited in the search for representative yet diverse features. In a GREEDY search, the next best feature to select is one that is a good representative of some unselected words, but also unlike previously selected words. This procedure maintains representativeness while ensuring diversity by discouraging redundant selections. Greedy search can of course result in locally optimal, yet globally non-optimal feature subsets. Therefore, a globally informed search, CLUSTER selects representative features from word clusters.

Central to feature selection methods introduced in this papers is the notion of similarity between words. Word co-occurrence behaviour is a good indicator of word similarity, however co-occurrence data derived from textual sources is typically sparse. Hence, distance measures must assign a distance to all word pairs, whether or not they co-occur in the data. Distributional similarity measures (obtained from information theory) achieve this by comparing co-occurrence behaviour on a separate disjoint set of target events [18]. In this paper events are all other words. Intuitively, if a group of words are distributed similarly with respect to other words then selecting a single representative from a neighbourhood of words will mainly eliminate redundant information. Consequently, this selection process will not hurt case representation, but will significantly reduce dimensionality. A further advantage of exploiting co-occurrence patterns is that it provides contextual information to resolve ambiguities in text such as similar meaning words that are used interchangeably (synonyms) and the same word being used with different meaning (polysemies). In both situations similar cases can be overlooked during retrieval if these semantic relationships are ignored.

Section 2 presents existing work in unsupervised feature selection and work related to distributional distance measures and clustering based indexing schemes. Next we establish our terminology before presenting the baseline method in Section 3. Details of distributional distance measures and the role of similarity for unsupervised feature selection is discussed in Section 4. Section 5 introduces the two similarity-based selection methods, GREEDY and CLUSTER. Experimental results are reported on four email datasets in Section 6, followed by conclusions in Section 7.

2 Related Work

Feature selection for structured data can be categorised into filter and wrapper methods. Filters are seen as data pre-processors and generally, unlike wrapper approaches, do not require feedback from the final learner. As a result they tend to be faster, scaling better to large datasets with thousands of dimensions, as typically encountered in text applications. Comparative studies in supervised feature selection for text have shown heuristics based on Information Gain (IG) and the Chi-squared statistic to consistently outperform less informative heuristics that rely only on word frequency counts [26].

Unlike with supervised methods, comparative studies into unsupervised feature selection are very rare. In fact, to our knowledge there has only been one publication explicitly dealing with unsupervised feature selection for text data [16]. Generally, existing unsupervised methods tend to rely on heuristics that are informed by word frequency counts over the text collection. Although frequency can be a fair indicator of feature utility it does not consider contextual information. Ignoring context can be detrimental for text processing tasks because ambiguities in text can often result in poor retrieval performance. A good example is when dealing with polysemous relationships such as "financial bank" and "river bank", where the word frequency for "bank" is clearly insufficient to establish its context and hence its suitability for indexing or case comparison.

In Textual Case-Based Reasoning (TCBR) research [22] the reasoning process can be seen to generally incorporate contextual information in two ways: as part of an elaborate indexing mechanism [2]; or as part of the case representation [24]. The latter requires simpler retrieval mechanisms, hence is a good choice for generic retrieval frameworks; while the former, although better at capturing domain-specific information, is more demanding of the retrieval process. A further distinguishing characteristic of TCBR systems is the different levels of knowledge sources employed to capture context [14]. These levels vary from deep syntactic parsing tools and manually acquired generative lexicons in the FACIT framework [7]; to semi-automated acquisition of domain-specific thesauri with the SMILE system; to automated clause extraction exploiting keyword co-occurrence patterns in PSI [25]. Of particular interest to this paper is the capture of co-occurrence based, contextual information within the case representation. Current research in this area is focused on feature extraction, which unlike feature selection aims to construct new features from existing features. Interest in this area has resulted in extraction techniques for both supervised (e.g. [25,27]) and unsupervised settings (e.g. [4,11]).

In text classification and applied linguistic research the problem of determining context is commonly handled by employing distributional clustering approaches. Introduced in the early nineties for automated thesaurus creation [18], distributional clustering has since been widely adopted for feature extraction with supervised tasks, such as text classification [1,20]. Word clusters are particularly useful because contextual information is made explicit by grouping together words that are suggestive of similar context. Additionally, word clusters also provide insight into vocabulary usage across the problem domain. Such information is essential if representative features are to be selected. Of particular importance for word clustering are distributional distance measures. These measures ascertain distance by comparison of word distributions

conditioned over a disjoint target set. Typically, class labels are the set of targets and so cannot be applied to unsupervised tasks.

The textual case retrieval system SOPHIA introduced a novel approach to combining distributional word clustering with textual case base indexing [17]. Here feature distance is measured by comparing word distributions conditioned on other co-occurring words (instead of class labels). Indexing is enabled by identifying seed features that act as case attractors. They argue that seed features are those that have non-uniform distributions having low entropy, referred to as specific word contexts. However the entropy based measure cannot distinguish between representative and diverse features even if they have specific contexts.

In structured CBR, clustering is commonly employed as a means to identify representative and diverse cases for casebase indexing. A good example is the footprint-driven approach [21] where a footprint case is: representative of its neighbourhood because of its influence; and diverse because its area of competence cannot be matched by any other case. This notion of identifying diverse yet representative cases has also been exploited in casebase maintenance [6,23].

In summary, the representativeness and diversity of an entity can be measured by analysing its neighbourhood. In this paper the entity is the feature and representativeness and diversity are also important for feature selection. Central to feature neighbourhood analysis is a good distance metric. When features are words, the distance metric must take context into account. Distributional distance measures do this by exploiting word co-occurrence behaviour.

3 Frequency Based Unsupervised Feature Selection

We first introduce the notation used in this paper to assist presentation of the different feature selection techniques. Let \mathcal{D} be the set of documents and \mathcal{W} the set of features, which are essentially words. A document d is represented by a feature vector, $\mathbf{x} = (x_1, \dots, x_{|\mathcal{W}|})$, of frequencies in d of words from \mathcal{W} [19]. In some applications, the frequency information is suppressed, in which case the x_i are binary values indicating the presence or absence of words in d . The main aim of unsupervised feature selection is to reduce $|\mathcal{W}|$ to a smaller feature subset size m by selecting features ranked according to some utility criterion. The selected m features then form a reduced word vocabulary set \mathcal{W}' , where $\mathcal{W}' \subset \mathcal{W}$ and $|\mathcal{W}'| \ll |\mathcal{W}|$. The new representation of document d is the reduced word vector \mathbf{x}' , which has length m .

Frequency counts are often used to gauge feature utility particularly in an unsupervised setting. The Term Contribution (Tc) is one such measure, showing promising results in [16]:

$$Tc(w) = \sum_{\substack{i,j \\ i \neq j}} F(w, d_i) * F(w, d_j)$$

$$F(w, d) = f(w, d) * \log_2 \frac{|\mathcal{D}|}{n}$$

Here F computes the $tf*idf$ score which is a measure of the discriminatory power of a word given a document. Term frequency f is the within document frequency count of a

feature and n is the number of documents containing feature w . Tc 's frequency based ranking and selection of features is the base line feature selection method used in this paper and we will refer to it as BASE (Figure 1).

```

m = feature subset size
BASE
  Foreach  $w_i \in \mathcal{W}$ 
    calculate  $Tc$  score using  $\mathcal{D}$ 
  sort  $\mathcal{W}$  in decreasing order of  $Tc$  scores
   $\mathcal{W}' = \{w_1, \dots, w_m\}$ 
  Return  $\mathcal{W}'$ 

```

Fig. 1. Feature selection with Tc based ranking

Tc will typically rank frequent words appearing in fewer documents above those appearing in a majority of documents. In this way the BASE method will attempt to ignore overly frequent (or rare) features. Its main drawback is its inability to address the need for both representative and diverse features. This leads to selection of non-optimal dimensions that fail to sufficiently capture the underlying document content.

4 Role of Similarity for Unsupervised Feature Selection

A representative feature subset is one that can discriminate between distinct groups of problem-solving situations. In a classification setting, these groups are identified by their class labels and are typically exploited by the feature selection process. However in the absence of class knowledge, we need to identify and incorporate other implicit sources of knowledge to guide the search for features.

Similar problem situations are typically described by a similar set of features forming an operational vocabulary subset. When these subsets are discovered the search for features can be guided by similarity in problem descriptions. In particular knowledge about feature similarity enables the search process to address both the need for representative and diverse features. The question then is how do we define similarity between features. A good starting point is to analyse feature co-occurrence patterns because features that are used together to describe problems are more likely to suggest the same operational vocabulary subset than features that rarely co-occur. In the rest of this section we look at how feature utility can be inferred from similarity knowledge extracted from feature co-occurrence patterns.

4.1 Feature Utility Measures

For a given word $w \in \mathcal{W}$, our first metric estimates the average pair-wise distance \overline{Dist} between w and its neighbourhood of k nearest word neighbours.

$$\overline{Dist}(w, \mathcal{A}, k) = \frac{1}{k} \sum_{w_N \in N_k(w, \mathcal{A})} Dist(w, w_N)$$

where N_k returns the k nearest neighbours of w chosen from $\mathcal{A} \subseteq \mathcal{W}$, and $Dist$ is the distance of w from its neighbour w_N . Lower values for \overline{Dist} suggests representative words that are centrally placed within dense neighbourhoods.

An obvious distance measure for words is to consider the number of times they co-occur in documents [19]. However the problem with such a straight forward co-occurrence count is that similar words can be mistaken as being dissimilar because they may not necessarily co-occur in the available document set \mathcal{D} . This is typical with text due to problems with sparseness [4].

4.2 Distributional Distance Measures

Often, related words do not co-occur in any document in a given collection, due to sparsity and synonymy. This limits the usefulness of similarity measures based purely on simple co-occurrence. Distributional distance measures circumvent this problem by carrying out a comparison based on co-occurrence with members of a separate disjoint target set [18]. Applied to text, the idea measures distances between word pairs by comparing their distributions conditioned over the set of other words. Since the conditioning is undertaken over a separate disjoint set, distances between non co-occurring word pairs need no longer remain unspecified.

Let us first demonstrate the intuition behind distributional distance measures by considering three words, a , b and c , and their fictitious word distribution profiles (see Figure 2). The x-axis contains a set of target events w_i , while the y-axis plots the conditional probabilities $p(w_i|w)$, for $w = a, b, c$. Comparison of the three conditional probability distributions suggests a higher similarity between a and b (compared to profiles of a and c). When target events on the x-axis are words, then a comparison between conditional probability distributions provides a similarity estimate based on word co-occurrence patterns. The next question then is how can we measure distance between feature distributions.

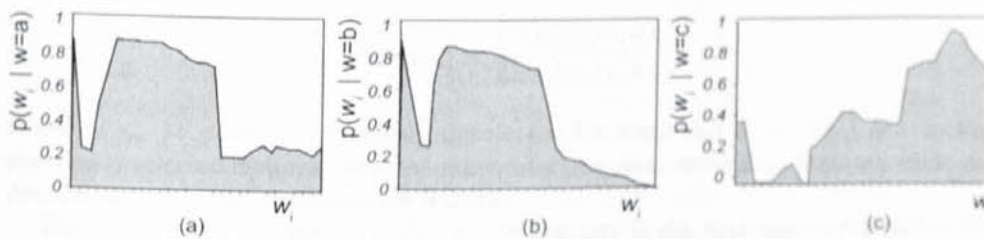


Fig. 2. Conditional probability distribution profiles

Let q and r be two features from \mathcal{W} whose similarity is to be determined. For notational simplicity we write $q(w_i)$ for $p(w_i|w = q)$ and $r(w_i)$ for $p(w_i|w = r)$, where $w_i \in \mathcal{W} \setminus \{q, r\}$ and p denotes probabilities calculated from the training data \mathcal{D} . Research in linguistics has shown that the α -Skew metric is a useful measure of the distance between word distributions, when applied to the task of identifying similar noun pairs [13]. It is argued that the asymmetric nature of this distance measure is appropriate for word comparisons, since one word (e.g. 'fruit') may be a better substitute

for another (e.g. 'apple') than vice-versa. Here we adopt this metric to compare word distributions and thereby determine the distance from word $q \in \mathcal{W}$ to word $r \in \mathcal{V}$.

$$Dist(q, r) = \sum_i r(w_i) \log \frac{r(w_i)}{q(w_i)}$$

is the Kullback-Leibler (KL) divergence, which is derived from information theory. It measures the average inefficiency in using $r(w_i)$ to code for $q(w_i)$ [3].

In our context, a large value of $Dist(q, r)$ would suggest that the word q is a poor representative of the word r , but not necessarily vice-versa. However, the $Dist$ is undefined if there are any words for which $q(w_i) = 0$, but $r(w_i) \neq 0$. The α -Skew metric avoids this problem by replacing q with $\alpha q + (1 - \alpha)r$, where the parameter α is less than one. In practice, our $Dist$ is the α -Skew metric with $\alpha = 0.99$, as suggested in [13].

5 Similarity Based Unsupervised Feature Selection Methods

\overline{Dist} is the simplest measure that can be employed to rank features. However, we wish to use it so that a diverse yet representative set of features is discovered. This can be achieved in two alternatively ways: a GREEDY search that is locally informed; or a more globally informed CLUSTER-based search.

5.1 Greedy Search for Features

What we propose here is a greedy local search for the best feature subset. At each stage, the next feature is selected to be both representative of unselected features and distant from previously selected features. The feature utility score FUS_k , combines the average neighbourhood distance \overline{Dist} from both the selected and unselected feature neighbourhoods as follows:

$$FUS_k(w) = \frac{\overline{Dist}(w, \mathcal{S}, k)}{\overline{Dist}(w, \mathcal{U}, k)}$$

where $\mathcal{U} \subseteq \mathcal{W}$ contains previously unselected features, and $\mathcal{S} = \mathcal{W} \setminus \mathcal{U}$ contains previously selected features. Here the numerator penalises redundant features while the denominator rewards representative features.

The FUS_k based ranking and selection of features is the first unsupervised feature selection method introduced in this paper and we will refer to it as GREEDY (Figure 3). Unlike Tc , FUS_k 's reliance on distributional distances to capture co-occurrence behaviour undoubtedly makes it far more computationally demanding. However this cost is justified by FUS_k 's attempt to address the need for both representative and diverse features. One problem though is that GREEDY is a hill-climbing search where the decision to select the next best feature is informed by local information, hence it can select feature subsets that, although locally optimal, can nevertheless be globally non-optimal.


```

 $m$  = feature subset size
 $S = \emptyset; \mathcal{U} = \mathcal{W}$ 
GREEDY
  Repeat
    For each  $w_i \in \mathcal{U}$ 
      calculate  $FUS_k$  score
    sort  $\mathcal{U}$  in decreasing order of  $FUS_k$  scores
     $w_j$  = top ranked feature in  $\mathcal{U}$ 
     $S = S \cup \{w_j\}$ 
     $\mathcal{U} = \mathcal{U} \setminus \{w_j\}$ 
  Until ( $|S| = m$ )
   $\mathcal{W}' = S$ 
  Return  $\mathcal{W}'$ 

```

Fig. 3. GREEDY method using FUS_k based ranking

5.2 Clustered Search for Features

Clustering of words provides a global view of word vocabulary usage in the problem description space. Each cluster contains words that are contextually more similar to each other than to words outwith their own cluster. Partitioning the feature space in this way facilitates the discovery of representative features because each cluster can now be treated as a distinct sub-part of the problem description space.

We use a hierarchical agglomerative (bottom-up) clustering technique, where at the beginning every feature forms a cluster of its own. The algorithm then unites features with greatest similarity in small clusters and these clusters are iteratively merged until m number of clusters are formed. The decision to merge clusters is based on the furthest neighbour principle, where those two clusters with least distance between their most dissimilar cluster members are merged. Typically, this form of cluster merging leads to tightly bound and balanced word clusters.

Merging of clusters requires that a distance metric is in place. For this purpose we use the *Dist* metric from Section 4. However, we must first address the question of how to deal with the asymmetrical nature of this metric when comparing distances between members of separate clusters. There are essentially three ways in which the two distances can be consolidated: use the maximum; the minimum or the average. We advocate the maximum distance, which combines with the furthest neighbour principle to form clusters in which there are no large distances.

Figures 4 and 5 illustrates how the choice of distances can affect the final cluster structure. In this example, clusters are formed with keywords extracted from a PC-Mac hardware FAQs mailing list. A closer look at the five word clusters formed using the maximum of the assymetrical distance between a feature-pair suggests that the resulting groups are not only semantically meaningful (e.g. cluster membership of "dos") but are also more balanced (e.g. number of words in a cluster).

Once clusters are formed we need a mechanism to uniformly select one representative feature from each cluster. In Figures 4 and 5 underlined words indicate such representatives (often referred to as cluster centroids or seeds). Previously, we stated that a

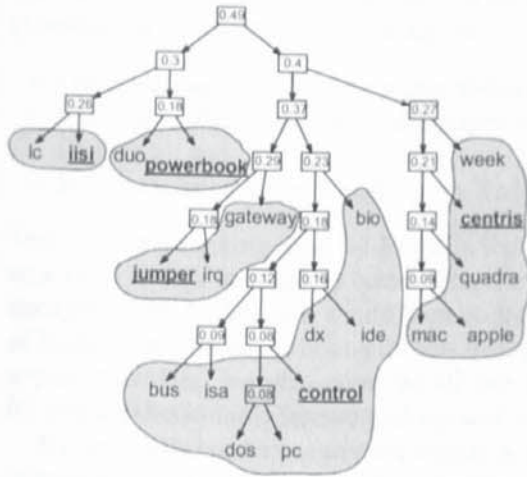


Fig. 4. Maximum distance



Fig. 5. Minimum distance

m = feature subset size
 $\mathcal{W}' = \emptyset$
 generate set of word clusters $\{C_1, \dots, C_m\}$
CLUSTER
 For each $C_i \in \mathcal{W}$
 w_j = feature with max FUS_C in C_i
 $\mathcal{W}' = \mathcal{W}' \cup w_j$
 Return \mathcal{W}'

Fig. 6. CLUSTER method using FUS_C based ranking

representative feature is identified by its placement within dense neighbourhoods. Using this same idea and the \overline{Dist} metric in Section 4 we estimate the representativeness of feature w within a cluster C as a function of the average pair-wise distance between itself and its cluster members:

$$FUS_C(w) = (1 - \overline{Dist}(w, C, |C|))$$

The second unsupervised feature selection method introduced in this paper is CLUSTER. It uses the FUS_C score to rank features in a cluster, choosing w with highest FUS_C from each cluster. The main steps appear in Figure 6. Here the number of clusters formed is equal to the desired feature subset size, m . This determines the stopping criterion for clustering. Like GREEDY, CLUSTER also addresses the need for representativeness and diversity, however, we expect CLUSTER to have an edge over GREEDY because its selection is influenced more globally.

6 Evaluation

We wish to determine the effectiveness of the two similarity-based searches for features, compared to the frequency-based search:

- GREEDY, introduced in this paper with ranking using FUS_k ¹ (Figure 3);
- CLUSTER, also introduced in this paper, exploits clustering and ranking using FUS_c (Figure 6); and
- BASE, the baseline with ranking on T_c (Figure 1).

The T_c -based ranking used by BASE is the only unsupervised method that has up to now been shown to perform better than the basic document frequency and the term strength methods [16]. We would hope to significantly improve upon the performance of BASE. Now the upper-bound for any unsupervised technique is its supervised counterpart, therefore, we also compare all our unsupervised methods with the standard IG-based SUPERVISED feature ranking and selection method.

It is generally harder to carry out empirical testing within a truly unsupervised setting compared to a supervised one. This is because, the absence of supervised labels calls for alternative sophisticated evaluation criteria, such as comparison of retrieval rankings or establishing measures of cluster quality. Instead, we applied our unsupervised methods on labelled data ignoring labels until the testing phase. Essentially we are exploiting class labels only as a means to evaluate retrieval performance which indirectly measures the effectiveness of the case representation. Note that we are not interested in producing a supervised classifier.

Experiments were conducted on 4 datasets; all involving email messages. Each email message belongs to one mail folder. Here folders are the class labels. As in previous experiments we used the 20Newsgroups corpus of 20 Usenet groups [9], with 1000 postings (of discussions, queries, comments etc.) per group, to create 3 sub-corpus [24]: SCIENCE (4 science related groups); REC (4 recreation related groups) and HW (2 hardware problem discussion groups, one on Mac, the other on PC). With each sub-corpus the groups were equally distributed. A further set of 1000 personal emails, used for Spam filtering research forms the final dataset, USREMAIL, of which 50% are Spam [5].

We created 15 equal-sized disjoint train-test splits. Each split contains 20% of the full dataset, selected randomly, but constrained to preserve the original class distribution. All text was pre-processed by removing stop words (common words) and punctuation and the remaining words were stemmed. In the interest of reducing time taken for repeated trials, the initial vocabulary size was cut down to a subset composed of the 500 most and 500 least discriminating words (using IG). These 1000 words then form \mathcal{W} . An effective feature selection method should eliminate the non-discriminating words and assemble a representative and non-redundant combination of the discriminating ones.

The effectiveness of feature selection is directly reflected by the usefulness of the case representation obtained. Therefore, case representations derived by GREEDY, CLUSTER, BASE and SUPERVISED are compared on test set accuracy from a retrieve-only system, where the weighted majority vote from the 3 best matching cases are used to

¹ In our experiments $k=15$ is used as FUS_k 's neighbourhood size.

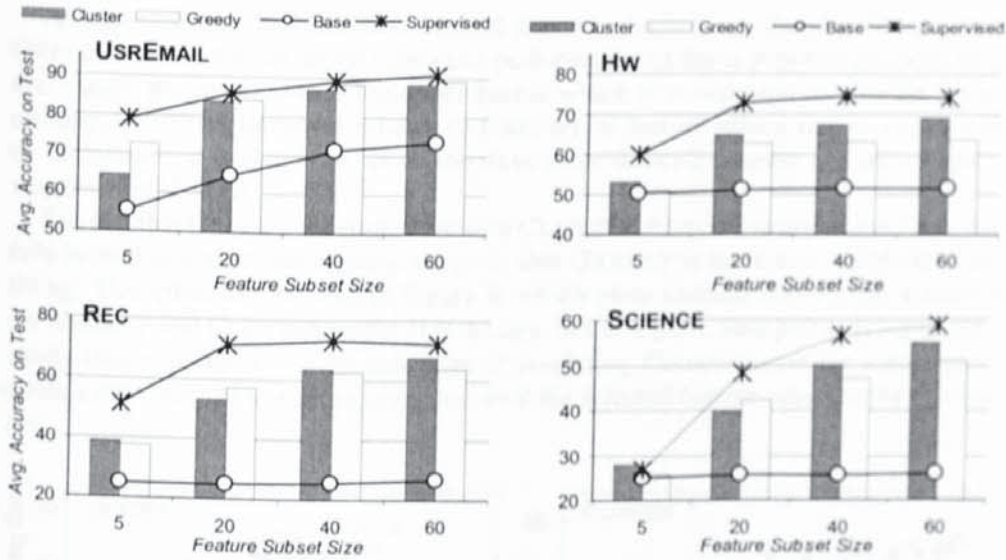


Fig. 7. kNN accuracy results for 4 datasets

classify the test case. For each test corpus and each method the graphs show the test set accuracy (averaged over 15 trials) computed for representations with 5, 20, 40 and 60 feature subset sizes (Figure 7).

6.1 Results

Analysis of overall performance of SUPERVISED on the 4 datasets indicates that the classification of emails from USREMAIL as Spam or legitimate presents the easiest task. Here, SUPERVISED obtained 80% accuracy with just 5 features, compared with only 60% accuracy on the SCIENCE dataset. In all datasets except SCIENCE, we observe a steep rise in accuracy up to about 20 features, followed by a levelling-off as more features are added. This indicates that SCIENCE is the most difficult problem. Unlike USREMAIL, the other binary-classed HW dataset is harder, because similar terminology (e.g. monitor, hard drive) can be used in reference to both classes (i.e. PC and Apple Mac). Additionally, the same hardware problem can be relevant to both mailing lists, resulting in cross-posting of the same message.

We note that BASE performs very poorly on all datasets compared to GREEDY, CLUSTER and SUPERVISED. With the exception of the easiest problem (USREMAIL), it barely outperforms random allocation of classes and does not improve its performance as more features are added. Both GREEDY and CLUSTER clearly outperform BASE on all four datasets and improve their performance as the number of features increase. BASE's poor performance is explained by the fact that it selects features purely on the basis of term frequency information. Although frequent words will co-occur with many other words these co-occurrences will not necessarily be with similar words. Since similar words are indicative of similar areas in the problem space, BASE is not able to identify words that are representative of the problem space.

As expected, the SUPERVISED method achieves highest accuracy. Although both GREEDY and CLUSTER never match the performance of the supervised method, they make good progress towards the upper bound which it is expected to provide. Interestingly, CLUSTER improves relative to GREEDY as feature subset size increases and by 60 features, it is clearly better on the three more difficult datasets and only slightly worse on USREMAIL.

The fact that GREEDY is competitive with CLUSTER at lower feature subset sizes, but falls behind at higher subset sizes, suggests that GREEDY is more susceptible to overfitting. This effect can be seen in Figure 8, which plots training and test set accuracy for GREEDY and CLUSTER on the HW dataset. In these plots, data points lying significantly above the line $x = y$ are indicative of overfitting. Comparison of the scatter-plots confirms that GREEDY is more likely to overfit the selected feature subset to the training set.

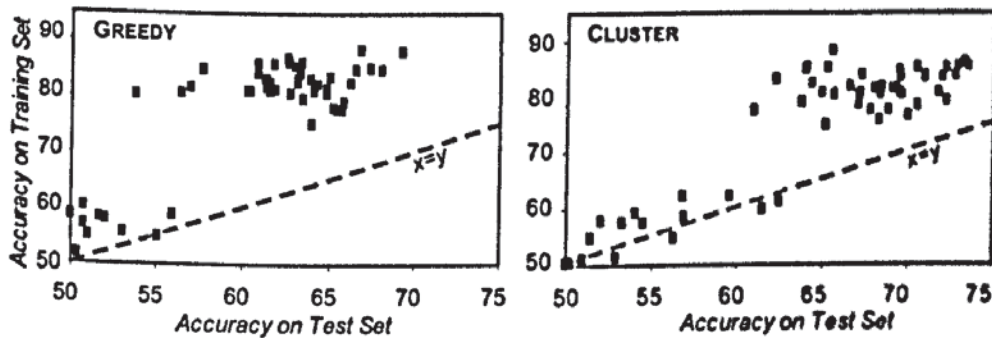


Fig. 8. Comparison of overfitting behaviour with GREEDY and CLUSTER on HW

6.2 Evaluation Summary

We checked the significance of observed differences between GREEDY and CLUSTER, using a 2-tailed t-test with a 95% confidence level for feature subset size, m equal to 60 (see Table 1). This test indicated that the superiority of CLUSTER over GREEDY was significant in all three datasets (bold font), but that of GREEDY on USREMAIL was not shown to be significant at this level. The superior scaling of CLUSTER can be explained by the fact that small optimal feature subsets need not be subsets of larger ones. GREEDY can be expected to suffer from overfitting at larger feature subset sizes, since the greedily chosen early features are locked in and cannot be altered to improve

Table 1. Results summary for feature subset size 60 according to significance

60 features	USREMAIL	HW	REC	SCIENCE
GREEDY	89.3	63.7	64.0	51.0
CLUSTER	88.3	69.1	67.7	54.9
BASE	73.5	51.7	26.5	26.2
SUPERVISED	90.8	74.0	72.0	58.7

the global quality of a larger feature set. CLUSTER avoids this problem by dividing the entire feature set into as many clusters as required, before then selecting one keyword to represent each cluster.

7 Conclusions

The methods introduced in this paper are particularly suited to generating case representations from free text data for unsupervised tasks. The novelty of these methods lies in their exploitation of distributional similarity knowledge to assess the utility of candidate features.

We introduce two unsupervised feature selection methods: GREEDY and CLUSTER. Key to both these methods is the selection of representative yet diverse features using similarity knowledge. Distributional distance measures are able to adequately capture feature similarity by addressing sparseness in co-occurrence data [18]. Evaluation results show significant retrieval gains with case representations derived by GREEDY and CLUSTER, over an existing proven method (BASE) from a previous comparative study [16]. It is also encouraging to report that both GREEDY and CLUSTER make good progress towards the upper bound which is provided by a standard supervised feature selection method. Generally GREEDY is able to generate good feature subsets early on in the search for features while CLUSTER's global search approach consistently outperforms the GREEDY search with increasing feature subset sizes. This is due to the locally informed GREEDY search identifying locally optimal, yet globally sub-optimal, subsets. Results also indicate that GREEDY is more susceptible to overfitting. We intend studying the influence of representativeness and diversity on overfitting, using a weighted form of FUS_C to control the balance between representativeness and diversity.

Previously we have shown that feature selection is a useful integral part of feature extraction when applied to text classification [24]. One difficulty that we have encountered since then, is that a majority of applications involving text are not necessarily supervised. This work is a first step towards resolving this shortcoming in existing feature discovery tools. Future work will look at combining feature selection with more powerful feature extraction methods to create comprehensive tools for text representation, indexing and retrieval for both supervised and unsupervised tasks.

References

1. Baker, L., McCallum, A.: Distributional clustering of words for text classification. In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval* ACM Press (1998) 96–103
2. Bruninghaus, S., Ashley, K.: The role of information extraction for textual CBR. In *Case-Based Reasoning Research and Development: Proceedings of the 4th International Conference on CBR* Springer (2001) 74–89
3. Cover, T., Thomas, J.: *Elements of Information Theory*. John Wiley (1991)
4. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41(6) (1990) 391–407

5. Delany, S., Cunningham, P.: An analysis of case-base editing in a spam filtering system. In *Proceedings of the 7th European Conference on Case-Based Reasoning* Springer (2004) 128–141
6. Delany, S., Cunningham, P., Doyle, D., Zamolotskikh, A.: Generating estimates of classification confidence for a case-based spam filter. In *Case-Based Reasoning Research and Development: Proceedings of the 6th International Conference on CBR* Springer (2005) 177–189
7. Gupta, K., Aha, D.: Towards acquiring case indexing taxonomies from text. In *Proceedings of the Seventeenth International FLAIRS Conference* AAAI Press (2004) 307–315
8. Jarmulak, J., Craw, S., Rowe, R.: Genetic algorithms to optimise CBR retrieval. In Enrico Blanzieri and Luigi Portinale, editors, *Proceedings of the 5th European Workshop on CBR* Springer (2000) 137–149
9. Joachims, T.: A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorisation. In *Proceedings of the Fourteenth International Conference on Machine Learning* (1997)
10. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning* (1994) 121–129
11. Kang, N., Domeniconi, C., Barbara, D.: Categorization and Keyword Identification of Unlabelled Documents. In *Proceedings of the 5th IEEE International Conference on Data Mining* (2005)
12. Lamontagne, L., Lapalme, G.: Textual reuse for email response. In *Proceedings of the 7th European Conference on Case-Based Reasoning* Springer (2004) 242–256
13. Lee, L.: On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics 2001* (2001) 65–72
14. Lenz, M.: Defining knowledge layers for textual CBR. In *Proceedings of the 4th European Workshop on CBR* Springer (1998) 298–309
15. David D. Lewis and Kimberly A. Knowles. Threading electronic mail: A preliminary study. *Information Processing and Management* 33(2) (1997) 209–217
16. Liu, T., Liu, S., Chen, Z., Ma, W.: An evaluation on feature selection for text clustering. In *Proceedings of the Twentieth International Conference on Machine Learning* (2003) 488–495
17. Patterson, D., Rooney, N., Dobrynin, V., Galushka, M.: Sophia: A novel approach for textual case-based reasoning. In *Proceedings of the Nineteenth IJCAI Conference* (2005) 1146–1153
18. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of english words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics* (1993) 183–190
19. Salton, G., McGill, M.: *An introduction to modern information retrieval*. McGraw-Hill (1983)
20. Slonim, N., Tishby, N.: The power of word clusters for text classification. In *Proceedings of the 23rd European Colloquium on Information Retrieval Research* (2001)
21. Smyth, B., McKenna, E.: Building compact competent case-bases. In Klaus-Dieter Althoff, Ralph Bergmann, and L. Karl Branting, editors, *Proceedings of the Second International Conference on Case-Based Reasoning* Springer (1999) 329–342
22. Weber, R., Ashley, K., Bruninghaus, S.: Textual case-based reasoning. *To appear in The Knowledge Engineering Review* (2006)
23. Wiratunga, N., Craw, S., Massie, S.: Index driven selective sampling for case-based reasoning. In *Case-Based Reasoning Research and Development: Proceedings of the 5th International Conference on CBR* Springer (2003) 637–651
24. Wiratunga, N., Koychev, I., Massie, S.: Feature selection and generalisation for textual retrieval. In *Proceedings of the 7th European Conference on Case-Based Reasoning* Springer (2004) 806–820

25. Wiratunga, N., Lothian, R., Chakraborty, S., Koychev, I.: Propositional approach to textual case indexing. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (2005)* 380–391
26. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorisation. In *Proceedings of the Fourteenth International Conference on Machine Learning (1997)* 412–420
27. Zelikovitz, S.: Mining for features to improve classification. In *Proceedings of Machine Learning, Models, Technologies and Applications (2003)*

Case Based Reasoning for Anomaly Report Processing

Nirmalie Wiratunga¹, Stewart Massie¹, Susan Craw¹, Alessandro Donati², and
Emmanuel Vicari²

¹ School of Computing
The Robert Gordon University
Aberdeen AB25 1HG, Scotland, UK
{nw|sm|smc}@comp.rgu.ac.uk

² European Space Agency
European Space Operations Centre
64293 Darmstadt, Germany

Abstract. Case representation challenges Textual Case-Based Reasoning (TCBR) system development. This is particularly true for unsupervised problem domains. We introduce a novel unsupervised feature extraction technique to derive a structured case representation from text data. Our approach analyses word co-occurrence patterns to calculate similarity between words and uses this similarity knowledge to select representative but diverse seed words. Sparse representations are avoided by learning to generalise seed words with feature extraction rules. Our approach is demonstrated on a TCBR prototype, CAM, developed as an initial step towards supporting the European Space Agency's (ESA) anomaly report processing task. ESA's reports are semi-structured, unsupervised documents composed mainly of free-text. CAM currently implements the retrieve stage of CBR, where the aim is to retrieve similar anomaly reports when presented with a new anomaly. An initial expert evaluation provides evidence to support CAM's retrieval.

1 Introduction

In this paper we look at the retrieval stage of the Case-Based Reasoning (CBR) cycle applied to the complex task of anomaly report matching for the European Space Agency (ESA). In particular we address the problem of deriving a structured case representation from unsupervised text data using feature selection and extraction techniques.

Case representation is a key design issue for the successful development of any CBR system. This is particularly true for a Textual CBR (TCBR) system which generally requires the application of feature selection or extraction techniques to reduce the dimensionality of the problem by removing non-discriminatory and sometimes detrimental features. Dimensionality reduction has been shown to be successful in improving accuracy and efficiency for supervised tasks in unstructured domains [20]. However, in an unsupervised setting feature selection/extraction is a far more challenging task because class knowledge is not available to evaluate alternative representations.

We introduce a novel feature selection technique where similarity between words is calculated by analysing word co-occurrence patterns followed by seed word selection

using a footprint-based feature selection method. A feature extraction technique using rules induced by Apriori is applied to generalise the seed words and reduce sparseness of the case representation. The technique is implemented in a prototype CBR Anomaly Matching demonstrator called CAM. Currently, CAM is able to retrieve similar reports when presented with a new anomaly and incorporates intuitive visualisation techniques to convey case similarity knowledge. An initial small-scale evaluation highlights the effectiveness of our approach.

The problem domain and the key objectives for the prototype are presented in Section 2. Section 3 discusses feature selection and extraction approaches used to create the case representation. We describe how the prototype was implemented in Section 4. Initial evaluation results are presented in Section 5 followed by related work in Section 6. Finally, we provide conclusions and recommendations for future work in Section 7.

2 Anomaly Reporting

ESA is Europe's gateway to space. Its mission is to shape the development of Europe's space capability and ensure that investment in space continues to deliver benefits to the citizens of Europe. ESOC, the European Space Operations Centre, is responsible for controlling ESA satellites in orbit and is situated in Darmstadt, Germany. ESOC works in an environment in which safety and Quality Assurance is of critical importance and, as a result, a formal Problem Management process is required to identify and manage problems that occur both within the operations of the space segment and of the ground segment. Observed incidents and problems (the cause of the incidents), are recorded by completing anomaly reports.

Anomaly reports are semi-structured documents containing both structured and unstructured data. There are 27 predefined structured fields containing information such as: the originator's name; key dates relating to the report and the physical location of the anomaly. Structured fields are used to group and sort reports, for example by urgency or criticality. Importantly, for knowledge reuse purposes the anomaly reports also have an observation (the title of the report), description (facts observed), recommendation (first suggestion on recovery) and resolution field (how the problem is analysed and disposed). These four unstructured fields contain free text that are not necessarily always spell-checked or grammatically perfect but contains valuable knowledge.

The Anomaly Report Tracking System (ARTS) is an application that supports ESA staff in tracking the anomaly reports from the creation to the final closure. ARTS ensures that reports have a standard format and allows the status of a report to be controlled as it progresses through the life cycle of a problem (recording, preliminary review, analysis, disposition, closure). There are approximately 1000 reports generated per year spanning multiple missions covering a variety of problems (e.g. operations incidents, software problems, non-conformance). Report content is entered by different people across different missions (or infrastructure entities) so report life cycle, level of detail and use of vocabulary varies, however some level of consistency in technical terms is to be expected. Although reports can originate from as far back as ten years, the reports in the ARTS system date back to just five years.

The work described in this paper involves the organisation and extraction of knowledge from anomaly reports maintained by the ARTS system. The overall goal is to extract knowledge and enable decision support by reusing past-experiences captured in these reports. An initial prototype CAM supports report linking and resolution retrieval.

- *Task 1:* Report linking aims to discover similar technical problems across multiple reports and to generate links between reports across projects. Reports can be related because they either describe symptoms of the same problem within the same project (indicating the re-occurrence of incidents associated with the same cause) or they report a similar anomaly shifted in time occurring in different projects / missions. Relating anomalies can highlight single problems that result in multiple incidents which are recorded in different (and sometimes un-related) reports. One goal is to find relationships in an automatic way.
- *Task 2:* Report reuse aims to retrieve similar reports so that their resolution can be re-applied to the current problem. This involves retrieval and reuse of anomaly reports with the requirement to compare new anomaly descriptions with past anomaly reports. In standard CBR terminology the resolution field provides the problem solution while the remaining fields decompose the problem description. Determining a suitable resolution for an anomaly is currently a manual decision making process (using Anomaly Review Boards) requiring considerable domain expertise. The prototype aids this decision-making process by providing the user with a list of anomaly reports that have similar problem descriptions to the current anomaly.

3 Document Representation

The first task for developing our prototype CBR system was to create a case representation for anomaly reports. The structured fields in the document were reduced to 13 relevant features following discussions with the domain experts.

Representation of the textual parts of the reports is a far harder task. The unstructured text has to be translated into a more structured representation of feature-value pairs. This involves identifying relevant features that belong to the problem space and solution space. The translation from text into a structured case representation can not be performed manually because the dimensionality of the problem is too great: there is a vocabulary of approximately 220,000 words in the training sample of 960 reports. An approach which can identify relevant features from the corpus is required. There are numerous approaches to feature selection and extraction on supervised problems where class knowledge can be used to guide the selection [11,20]. However since we are faced with an unsupervised problem the selection needs to be guided by knowledge other than class. For this purpose we exploit word similarity knowledge.

Our approach to unsupervised feature extraction consists of three stages: an initial vocabulary reduction by pre-processing text using standard IR and NLP techniques; next seed word selection using word distribution profiling; and finally feature extraction using Apriori rules to avoid sparse representations.

3.1 Initial Dimensionality Reduction

The initial vocabulary of 220,000 words is reduced to 2500 words by applying the following document pre-processing techniques:

- Part of Speech Removal: text is first tokenised to identify word entities then tagged by its part of speech. Only nouns and verbs are retained.
- Stop Word Removal and Stemming: removes commonly occurring words and reduces remaining words to their stem by removing different endings, e.g., both anomaly and anomalous are stemmed to their root anomaly.
- Frequency Based Pruning: reduces the vocabulary, from approximately 8000 words to 2500 words, by considering the inverse document frequency (idf) of each word to determine how common the word is in all of the documents. Typically we accept words with an idf value of between 3 and 6.

3.2 Seed Word Selection by Word Profiling

Our aim in this part of the document representation process is to select words that are representative of areas of the problem space but that are also diverse so that together they provide good coverage of the problem space. Knowledge about word similarity enables the search process to address both these requirements. The question then is how do we define similarity between words and thereafter how do we select representative but diverse words.

Word Similarity: One approach is to consider the number of times words co-occur in documents [14], however, a problem is that similar words do not necessarily co-occur in any document, due to sparsity and synonymy, and will not be identified as similar.

Our approach is to analyse word co-occurrence patterns with the set of words contained in the solution, i.e., the remaining words from the resolution field. For example, to calculate the similarity between words in the observation field of the anomaly report, the conditional probability of co-occurrence is first calculated between each word in the observation field with each word in the resolution field. A distribution of these probabilities is then created for each observation word. A comparison between these distributions can then be made using the α -Skew metric derived from information theory [10]. This comparison provides an asymmetric similarity estimate between words in the observation field. We repeat the same process for all the text fields. Essentially similar words are those that have similar co-occurrence patterns with resolution words. A full description of this word similarity approach is given in [19].

Representative but Diverse Selection: We use the similarity knowledge derived from the conditional probability distributions to aid the search for a representative but diverse set of seed words. These words form the dimensions for the case representation. Smyth & McKenna developed a footprint-based retrieval technique in which a subset of the case base, called footprints, is identified to aid case retrieval [16]. We use a similar technique to cluster words and then select representative seed words from word clusters.

Word clusters are created by first forming coverage and reachability sets for each word. In our scenario, the coverage set of a word contains all words within a predefined similarity threshold. Conversely, the reachability set of a word is the set of words that contains this word in its coverage set. Clusters of words are then formed using the reachability and coverage sets to group words that have overlapping sets. In Figure 1, six words (w_1 to w_6) are shown spaced in relation to their similarity to each other. The coverage of each word is shown by a circle with a radius corresponding to the similarity threshold. It can be seen that two clusters are formed: w_1 to w_5 in one cluster and w_6 in the other. A representative set of seed words is selected for each cluster by first ranking the words in descending order of relative coverage [16]. Each word is then considered in turn and only selected if it is not covered by another already selected word. In Figure 1 the words are shown, in ranked order, with their coverage sets and related coverage scores. Hence w_1 , w_5 and w_6 will be selected as the seed words. The composition of the coverage sets depends upon the similarity threshold chosen and so the number of seed words formed can be varied by adjusting this threshold.

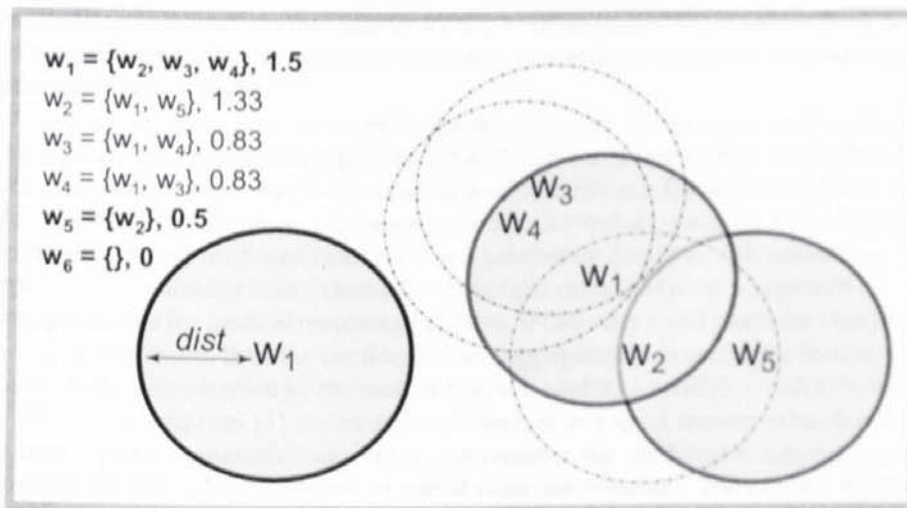


Fig. 1. Seed word selection using the footprint technique

3.3 Feature Extraction using Apriori Rules

Each seed word forms a feature in the case representation. A feature value is derived based on the presence of a seed word in the report. Using seed words alone in this way to represent free text results in a sparse representation. This is because reports may still be similar even though they may not contain seed words. One way around this problem is to embed the context of the seed within the case. We achieve this by the induction of feature extraction rules [18].

Feedback for Seed Word	Context Relevance					Expert's Comments
	1	2	3	4	5	
TLM						
TLM <= lost				x		Also a type of problem about telemetry: we loose it!
TLM <= process & receive		x				We receive it and we process it. These are actions
TLM <= lock					x	Bingo: We sometimes have "Telemetry lock problems"
TLM <= telemetry						It is the abbreviation / synonym
TLM <= available & generated	x					a bit weird
TLM <= product & generated		x				A more elaborated processing of the telemetry

Fig. 2. Feature extraction rules concluding the seed word, TLM.

Each rule associates words with a selected seed word, such that the rule conclusion contains the seed word and the rule body (or conditions) consists of associated words. The presence of associated words in a report (in the absence of the seed word) activates the rule, inferring a degree of seed word presence in the report. The mechanism of translating rule activations into feature values involves combining evidence from multiple rule activations. Essentially with increasing rule activations the problem with sparse representation decreases.

A rule's accuracy is reflected by its confidence score. When a rule with high confidence is activated it suggests higher belief in the presence of the seed word. Therefore, an activated rule's confidence score can be used to arrive at a feature value. Since a single seed word can be inferred from multiple rule activations, we need a mechanism to aggregate, the degree of seed word presence, inferred by these multiple activations. One approach to combining belief values from multiple rules is Mycine's approach to combining evidence for medical reasoning [4]. Here, if two rules x and y activate concluding the same seed word, then the confidences are aggregated to generate the feature value for the feature represented by the seed as follows: $\text{conf}(x) + \text{conf}(y) - \text{conf}(x) * \text{conf}(y)$

We use the Apriori [1] association rule learner to extract feature extraction rules. Apriori typically generates many rules, and requires that confidence, support and discriminatory thresholds be set before useful rules are generated. Here expert feedback on the quality of generated rules is vital. The explicit nature of rules is an obvious advantage both to establish context and also to acquire expert feedback (see Figure 2).

4 The CAM Prototype

Our CAM demonstrator uses the structured representation of the anomaly reports, created by the feature extraction process described in Section 3, as its case base. The representation process provides a 5 part case representation for each report. One of these contains the 13 features from the original structured report fields, while the remaining four parts are representations of the text data in the observation, description, recommendation and resolution fields of the report. These are represented by 75, 54, 80, and 150 features respectively, and correspond to the number of seed words extracted by the footprint-based feature selection. The similarity threshold controlling this extraction

was set to encourage balanced word clusters. We are currently working with a sample of 960 reports, supplied by ESA, for training purposes.

The retrieval strategy implemented on CAM uses the k Nearest Neighbour algorithm (k -NN) with feature weighting to identify the k most similar cases to the current problem. The relative importance of each field (1 structured + 4 text) in relation to identifying similar anomaly reports is established by setting a weight for each field. Feature weights can also be set within each field to establish the importance of individual features. A common approach for setting retrieval weights is to learn feature importance from the available cases [7]. However, this is difficult in an unsupervised setting and is currently a manual process.

CAM provides an interface (see Figure 3) that displays the current report at the top with a ranked list of similar, retrieved reports below together with their similarity scores. Both the current or retrieved reports can be viewed as a combined or as a field-based representation by selecting the tab for the appropriate pane. The structured representation is the default report view, however, alternative views display the original text. The seed words selected are used in the structured representation to label the features and the relevant feature extraction rules are also accessible by the user.

Two visualisations are available to assist the user compare similarities and differences between retrieved reports. A parallel co-ordinate plot, displayed on the bottom left of Figure 3, shows the similarity of the retrieved nearest neighbours to the current report. The similarities are shown for the overall representation and individually for each of the five fields. This visual representation of similarity knowledge is useful to explain retrieval results to the user. A second visualisation, displayed on the bottom right of Figure 3, uses the spring-embedder model [9] to preserve the similarity relationship between cases as on-screen distances. This visualisation provides a 2-dimensional view of the case-base that allows clusters of cases to be identified.

5 Experimental Evaluation

It is generally accepted that evaluation is a challenge for TCBR systems. Standard IR systems advocate precision and recall based evaluation on tagged corpuses. The manual tagging involves not only class assignment but often assignment of relevance judgements on retrieved sets. In practical situations it is clearly prohibitive to expect a domain expert to tag substantial numbers of cases with relevance judgements. A more reasonable approach is to acquire qualitative feedback on a few selected test cases.

In order to evaluate the CAM prototype, 5 probe reports were randomly selected from the sample of 960. For each probe the 3 most similar reports retrieved by CAM were noted. A further 3 randomly selected reports were then added to create a retrieval set size of 6. Each probe and the corresponding retrieval set was then presented to the domain expert. Importantly the domain expert was unaware about the source of the retrieval set. A standard questionnaire was then used to obtain our expert's feedback for each probe.

Figure 4 shows a snippet of the questionnaire. The graph in the figure, summarises the qualitative feedback received for all 5 probes. It is clear that CAM's median values are significantly higher across the 5 probes (Mann-Whitney $p=0.0001$). CAM's μ and σ

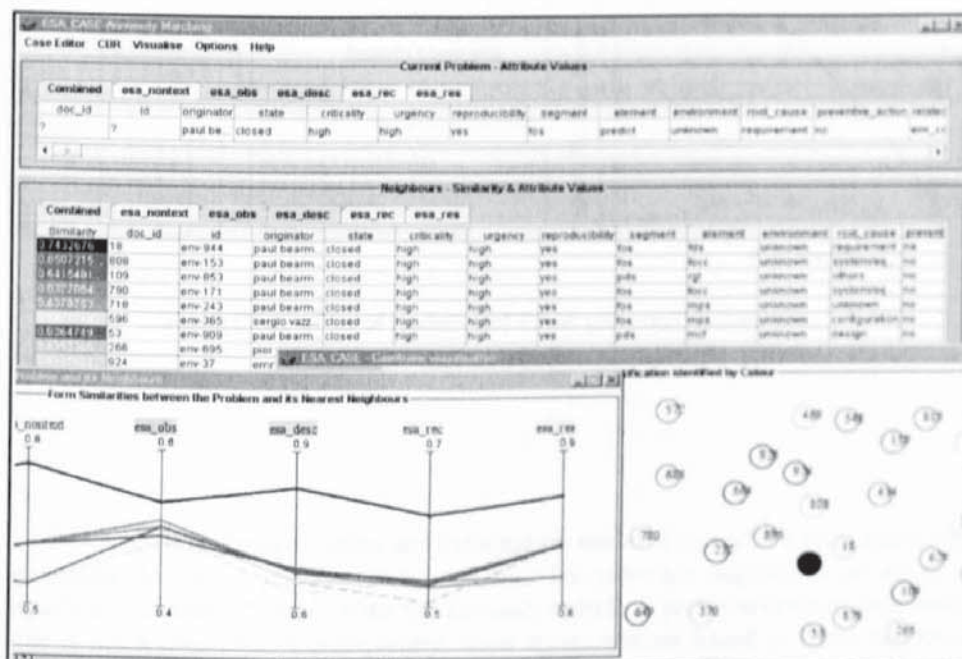


Fig. 3. Screen shot of CAM's interface

performance with probe 5 is due to the lack of text in the description, recommendation and resolution fields. With each of the remaining probes, the CAM generated retrieval set achieved the highest score of 4 for at least one of the cases in the set and a score of 3 with the other.

6 Related Work

A common problem for TCBR system development is the demand on knowledge acquisition. For instance in the EXPERIENCEBOOK project (aimed at supporting computer system administrators) all knowledge was acquired manually. This is not an exception, because current practice in TCBR system development show that the indexing vocabulary and similarity knowledge containers are typically acquired manually [17]. Consequently maintenance remains a problem since these systems are not able to evolve with newer experiences. These difficulties have created the need for fully or semi automated extraction tools for TCBR.

Tools such as stemming, stop word removal and domain specific dictionary acquisition are frequently used to pre-process text and are mostly automated. Acquiring knowledge about semantic relationships between words or phrases is important but is harder to automate. Although NLP tools can be applied they are often too brittle partly because they tend to analyse text from a purely linguistic point of view. Furthermore the reliance on deep syntactic parsing and knowledge in the form of generative lexicons still warrants significant manual intervention [6].

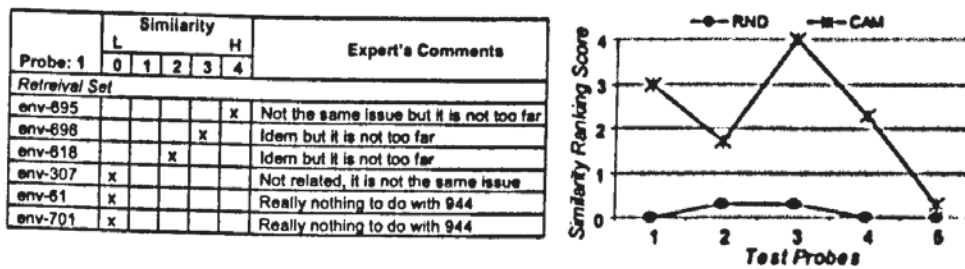


Fig. 4. Comparison of random and CAM generated retrieval sets.

Research in text classification and information retrieval typically adopts statistical approaches to feature selection and extraction. The main pre-requisite is access to a significant number of cases. With the anomaly reporting problem domain case base size is not a constraint. Consequently, word co-occurrence based analysis becomes particularly attractive for automated indexed vocabulary acquisition. A common approach to determining representative features involve the use of distributional clustering approaches [13], and has since been adopted for feature extraction with supervised tasks [15, 2]. Of particular importance for word clustering are distributional distance measures. These measures ascertain distance by comparison of word distributions conditioned over a disjoint target set. Typically, class labels are the set of targets and so cannot be applied to unsupervised tasks. However, in the SOPHIA retrieval system reliance on class labels was dropped by comparing word distributions conditioned on other co-occurring words (instead of class labels) [12]. Unlike with anomaly reports, SOPHIA operates on IR like documents, hence there is no requirement to learn from the differences between solution and problem space vocabulary. Our approach to calculating distributional distances is novel in that words from the problem space are compared conditioned on the solution space. This creates a distance measure that is guided by both the similarity and differences between problem and solution vocabularies.

Formation of newer and improved dimensions for case representation fall under feature extraction research. LSI is a popular dimensionality reduction technique particularly for text. Extracted features are linear combinations of the original features which unfortunately lack in expressive power [5]. Modelling keyword relationships as rules is a more successful strategy that is both effective and remains expressive. A good example is RIPPER [3], which adopts complex optimisation heuristics to learn propositional clauses for classification. Unlike RIPPER rules, association rules do not rely on class information and incorporates data structures that are able to generate rules efficiently making them ideal for large scale applications [21, 8]. The seed generalisation approach discussed in this paper is similar to that employed by the PSI tool introduced in [18], but unlike PSI here generalisation does not rely on class knowledge.

7 Conclusions and Future Work

The paper presents an initial approach to case retrieval applied to anomaly reports. It is a first step towards developing a CBR system to support the ESA's anomaly report processing task. Like most text applications, anomaly processing is unsupervised. It requires automated knowledge acquisition tools that are not reliant on class knowledge.

The paper introduces a novel unsupervised index vocabulary acquisition mechanism to map unstructured parts of text data to a structured case representation. For this purpose word pair-wise distances are calculated according to similarity in co-occurrence patterns over the solution space. This facilitates problem space words to be considered similar with specific reference to the solution space vocabulary.

Seed words are identified using word clusters and forms the features vector for the case representation. The idea of using a footprint-based feature selection strategy is novel. It facilitates selection of representative and diverse words but importantly does not require that the number of seed words be pre-specified. It does however require a similarity threshold to be in place which directly controls the feature vector size.

Presented techniques are implemented in the CAM prototype. Initial results from a small-scale qualitative evaluation of CAM's retrieval sets shows significant qualitative improvements over random retrieval. However we have yet to establish a principled approach to setting Apriori's parameters and the similarity threshold for the feature vector size. We expect that this will require better use of word similarity knowledge. Future work will extend CAM for the reuse and revision stages of the CBR cycle. Importantly more effort will be invested on evaluating all stages of the CBR cycle.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.: Fast discovery of association rules. In *Advances in Knowledge Discovery and DM* 307–327 AAAI Press (1995)
2. Baker, L., McCallum, A.: Distributional clustering of words for text classification. In *Proceedings of the 21st ACM Int Conf on IR* 96–103 ACM Press (1998)
3. Cohen, W., Singer, Y.: Context-sensitive learning methods for text categorisation. *ACM Transactions in Information Systems* 17(2):141–173 1999
4. Davis, R., Buchanan B., and Shortliffe, E.: *Production Rules as a Representation for a Knowledge-Based Consultation Program*. *Artificial Intelligence* 8 15–45 (1977)
5. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: Indexing by latent semantic analysis. *American Society of Information Science* 41(6) (1990) 391–407
6. Gupta, K., Aha, D.: Towards acquiring case indexing taxonomies from text. In *Proceedings of the 7th Int FLAIRS Conf* 307–315 AAAI Press (2004)
7. Jarmulak, J., Craw, S., Rowe, R.: Genetic algorithms to optimise CBR retrieval. In *Proc of the 5th European Workshop on CBR* 137–149 Springer (2000)
8. Kang, N., Domeniconi, C., Barbara, D.: Categorization and keyword identification of unlabelled documents. In *Proceedings of the 5th IEEE Int Conf on Data Mining* (2005)
9. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. *Information Processing Letters* 31(1) 7–15 (1989)
10. Lee, L.: On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics 2001* 65–72 (2001)

11. Liu, T., Liu, S., Chen, Z., Ma, W.: An evaluation on feature selection for text clustering. In *Proc of the 12th Int Conf on ML* 488–495 (2003)
12. Patterson, D., Rooney, N., Dobrynin, V., Galushka, M.: Sophia: A novel approach for textual case-based reasoning. In *Proc of the 19th IJCAI Conference* (2005) 1146–1153
13. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of English words. In *Proc of the 30th Annual Meeting of the Association for Computational Linguistics* 183–190 (1993)
14. Salton, G., McGill, M.: *An introduction to modern IR*. McGraw-Hill (1983)
15. Slonim, N., Tishby, N.: The power of word clusters for text classification. In *Proc of the 23rd European Colloquium on IR Research* (2001)
16. Smyth, B., McKenna, E.: Footprint-based Retrieval. In *Proc of the 3rd Int Conf on CBR* Springer 343–357 (1999)
17. Weber, R., Ashley, K., Bruninghaus, S.: Textual case-based reasoning. *To appear in The Knowledge Engineering Review* (2006)
18. Wiratunga, N., Lothian, R., Chakraborty, S., Koychev, I.: Propositional approach to textual case indexing. In *Proc of the 9th European PKDD Conf* (2005) 380–391
19. Wiratunga, N., Lothian, R., Massie, S.: Unsupervised Feature Selection for Text Data. In *Proc of the 8th European Conf CBR* Springer (2006)
20. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorisation. In *Proc of the 14th Int Conf on ML* (1997) 412–420
21. Zelikovitz, S.: Mining for features to improve classification. In *Proc of ML Models, Technologies and Applications* (2003)

Complexity-Guided Case Discovery for Case Based Reasoning

Stewart Massie and Susan Crow and Nirmalie Wiratunga

The Robert Gordon University
Aberdeen, AB25 1HG, Scotland, UK

sm@comp.rgu.ac.uk smc@comp.rgu.ac.uk nw@comp.rgu.ac.uk

Abstract

The distribution of cases in the case base is critical to the performance of a Case Based Reasoning system. The case author is given little support in the positioning of new cases during the development stage of a case base. In this paper we argue that classification boundaries represent important regions of the problem space. They are used to identify locations where new cases should be acquired. We introduce two complexity-guided algorithms which use a local complexity measure and boundary identification techniques to actively discover cases close to boundaries. The ability of these algorithms to discover new cases that significantly improve the accuracy of case bases is demonstrated on five public domain classification datasets.

Introduction

Case Based Reasoning (CBR) solves new problems by re-using the solution of previously solved problems. The case base is the main source of knowledge in a CBR system and, hence, the availability of cases is crucial to a system's performance. It is the availability of cases that often supports the choice of CBR for problem-solving tasks, however, in real environments there are often gaps in the coverage of the case base because it is difficult to obtain a collection of cases to cover all problem-solving situations.

Adaptation knowledge can be used to provide solutions to new problems that occur in the gaps that result from a lack of case coverage. However, gaining effective adaptation knowledge may be impossible or require considerable knowledge acquisition effort. The inclusion of additional, strategically placed cases can provide a more cost-effective solution.

Case discovery is the process of identifying *useful* new cases to fill gaps that exist in the coverage of the case base. This is different from traditional case learning, through the retain stage of the CBR cycle, in which newly solved problems are routinely added to the case base to assist future problem-solving. Rather case discovery is an active learning problem in which the aim is to identify areas of the problem space in which new cases would help to improve the system's performance and to create cases to fill these gaps. Commercial systems generally assume that a suitable case

base already exists and give little help to the case author in the case discovery stage of building the case base. There is a need for techniques to assist the case author during this crucial case base development stage.

We argue that new cases should be placed in regions of the problem-space in which the system is uncertain of the solution, and that these regions are generally close to boundaries between classifications. In this paper we present a new technique to identify and rank these areas of uncertainty and create candidate cases to assist the case author place new cases in these regions.

The remainder of this paper describes our approach and evaluates it on several public domain case bases. The next section discusses existing work on case discovery. The following sections outline how we use a complexity metric, boundary detection and clustering to identify areas of the problem-space that need the support of new cases and how these cases are created. The approach is then evaluated against two benchmark algorithms before we draw some final conclusions.

Related Work In Case Discovery

The case discovery problem can be considered in two stages. First *interesting* areas or gaps within the coverage of the case base must be identified and secondly cases must be created to fill these gaps. This presents a more complex challenge when compared to the more commonly researched case base editing or selective sampling problems that have a pool of existing cases from which to select cases. In contrast, the task of case discovery is to add to the case knowledge using implicit information held within the case base.

Some research has focused on the first stage of the discovery process. One approach to identifying gaps has been to focus on locating maximal empty hyper-rectangles within k -dimensional space (Liu, Ku, & Hsu 1997). In their later research the algorithm is capable of locating hyper-rectangles within data containing both continuous and discrete valued attributes (Liu *et al.* 1998). The main problem with this approach is that there is no way to identify if the gap found in the problem space is *interesting* from a problem-solving view-point, or even represents a possible combination of attribute values. An alternative approach to identifying interesting areas for new cases is proposed in (Wiratunga, Crow, & Massie 2003) as part of a selective sampling technique.

In this approach the case base plus unlabelled examples are formed into clusters using a decision tree technique. The clusters are then ranked based on the mixture of classes present, and then unlabelled examples are ranked based on their distance from labelled cases and closeness to other unlabelled examples. However this approach draws on knowledge gained from a pool of unlabelled cases not normally available during the case discovery process.

CaseMaker (McSherry 2001) is a knowledge acquisition tool that addresses both stages of the discovery process. A complete set of all uncovered cases is found by exhaustively searching the space of allowable feature combinations. These cases are ranked based on their potential coverage contributions, calculated using an adaptation function based on feature differences. This approach has been shown to be successful in finite domains where suitable domain knowledge and adaptation knowledge are available.

Competence-guided case discovery is an alternative approach based on a competence model (McKenna & Smyth 2001). The model groups together cases that solve each other into clusters called competence groups (Smyth & McKenna 2001). For each pair of competence groups the two closest cases are identified as the boundary pair and gaps are identified as the space between these *nearest neighbour* competence groups. Starting with the smallest gap a case is created whose feature values are determined by the cases in the neighbourhood of the boundary pair. While this intuitive approach is likely to discover valid cases in active regions of the problem space it gives no guarantee on finding the *most interesting* gap as it ignores large parts of the problem space.

Exploiting the existing knowledge within the case base is common to all the approaches discussed here and, likewise, we use this knowledge source to identify areas of uncertainty within the problem space and then to identify cases on classification boundaries.

Complexity-Guided Case Discovery

Our aim is to discover cases that improve the CBR system's accuracy. We believe cases close to classification boundaries are most likely to achieve this aim. As discussed earlier, the case discovery problem can be considered in two stages: identification of interesting areas of the problem space in which to place new cases is discussed in this section while the creation of new cases to fill these gaps is discussed in the following section.

Previous research on case base editing has highlighted the importance of cases in boundary regions for the competence of a case base (Brighton & Mellish 2002; Wilson & Martinez 2000). It seems reasonable to expect a successful case creation algorithm to also identify cases on class boundaries. Our approach to identifying where new cases should be placed, in order to improve a system's accuracy, involves several stages that combine to identify boundary cases.

Areas of Uncertainty The first stage in finding interesting areas for new cases is to find areas in which cases are likely to be wrongly classified. We do this by using a local complexity metric.

Classification complexity is an inherent problem characteristic that gives a measure of how difficult it is to classify new problems. It is determined by such factors as the overlap and length of decision boundaries, the dimensionality of the feature space, the noise level and the sparseness of available cases. Accuracy gives one measure of complexity but is dependent on the classifier chosen and provides no local information on areas of complexity within the problem space.

Several approaches have been used to estimate overall classification complexity. However, in case discovery we are interested in the complexity at local areas. We have chosen an approach that allows us to measure the local complexity based on the spatial distribution of cases rather than on a probabilistic distribution. In this approach the complexity of each case is calculated using a metric based on its *k*-Nearest Neighbours while incrementally increasing the value of *k*.

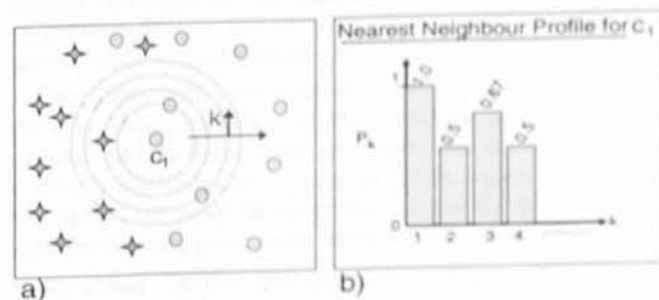


Figure 1: Complexity metric calculation

The complexity measure is calculated for each case by looking at the class distribution of cases within its local neighbourhood. P_k is the proportion of cases within a case's *k* nearest neighbours that belong to the same class as itself. In Figure 1a, as the value of *k* increases, the sequence of P_k starts 1, 0.5, 0.67, 0.5. A nearest neighbour profile can now be plotted of P_k as *k* increases. The complexity metric used is the area of the graph under the profile with the *x*-axis normalised, shown by the shaded area on Figure 1b. Case complexity is calculated by

$$\text{complexity} = 1 - \frac{1}{K} \sum_{k=1}^K P_k$$

for some chosen *K*. With *K*=4 the complexity of c_1 is 0.33. As the metric is weighted to a case's nearest neighbours using a large value for *K* has little impact on the results and *K*=10 was used in our calculations.

Cases with high complexity are close to classification boundaries and identify areas of uncertainty within the problem space. The regions around these target cases are identified as requiring support. Target cases are ranked in descending order of complexity to prioritise between the different regions.

Class Boundaries The case complexity metric is used to identify target cases in regions of the problem space near classification boundaries that we believe would benefit from the support of additional cases. However, it gives no help on

where, within these regions, the new cases should be placed. Following our hypothesis that cases close to class boundaries are important in case discovery we want to discover cases closer to the boundaries. There must be a classification boundary in the problem space close to the target case, however its direction and location are not known. To find an outer limit for the location of the boundary the target case's nearest unlike neighbour (NUN) is found i.e. the nearest case that has a different class. The boundary lies between these two reference cases i.e. the target case and its NUN.

Clustering Prioritising regions of the problem space using only the complexity value of cases is expected to identify interesting areas in which additional cases will improve the system's accuracy. However, prioritising on case complexity alone potentially gives two problems. There is a danger that new cases will be concentrated in a small region of the problem space as high complexity cases are likely to be located close to each other. In addition, new cases may be concentrated on small pockets of cases whose classification is different to their neighbours, as these cases will have high complexity values, resulting in poorer performance in noisy or multi-class problems.

Partitioning the case base into clusters may give a more balanced distribution of discovered cases over the whole case base. Competence group clustering (Smyth & Keane 1995) is a commonly used clustering technique in CBR and a similar approach has been adopted here. Clusters are formed using leave-one-out testing to measure the problem-solving ability of a case using: coverage and reachability. Coverage of a case is the set of problems that case can solve; conversely, reachability is the set of all cases that can solve it. Next clusters of cases, called competence groups, are formed using their reachability and coverage sets to group cases that have overlapping sets. This clustering model is typically applied to CBR systems incorporating an adaptation stage, however, here it is being applied to retrieve-only classification. In this scenario, the reachability set of a case is its k-nearest neighbours with the same classification but bound to the first case of a different class (Brighton & Mellish 1999).

With the case base formed into clusters, the complexity of each cluster can be defined as the average complexity of the cases it contains. The clusters can be ranked in descending order of complexity. Now, rather than choosing target cases purely on complexity ranking, one case can be chosen from each cluster with cluster complexity used to prioritise the target cases. The target case chosen from each cluster is the case with the highest complexity. In addition, there is now the opportunity to remodel the case base, by reforming the clusters as new cases are added, and building the effect of the discovered cases into the next round of case discovery.

Creating a New Case

The methods described in the previous section are used to identify interesting areas of the problem space for new cases. The second stage of the case-discovery process is to create a *candidate* case to occupy the area between the two

reference cases. This involves setting suitable feature values for the candidate case.

Candidate Case Feature Values Two approaches for setting the candidate case's feature values were investigated. In the first, the feature values are set as either the mean (numeric features) or majority (nominal features) of the feature values of the reference cases and their related sets. A case's related set is the union of its coverage and reachability sets. This approach, used by McKenna & Smyth, was found not to work well in domains containing small groups of exceptional cases. This may be due to one of the reference cases coming from a much larger competence group and applying excessive influence on the feature values, and hence location, of the candidate case. An alternative simpler approach was found to give more consistent results and was adopted for the complexity-guided algorithms. In this simpler approach the candidate case uses only the boundary pair to set its problem feature values. This results in a discovered case more evenly spaced between the reference pair.

Case discovery aims to create a new case for inclusion in the case base. Inclusion of the candidate case may be automatic but, as there is no guarantee that a candidate case will be a valid case occupying an active region of the problem space, the more likely scenario is for the case author to validate the case prior to its inclusion in the case base.

Noise Filter A potential problem of discovering cases on classification boundaries is that noisy cases may be discovered in domains containing significant levels of noise or exceptional cases. Indeed, most modern case editing algorithms (Brighton & Mellish 2002; Delany & Cunningham 2004) apply noise reduction algorithms prior to an editing approach that retains boundary cases.

A typical approach to noise reduction is to remove cases that are incorrectly classified (Wilson 1972). We apply a similar approach to determine if a validated case should be included in the case base. A *friend to enemy* distance ratio is calculated using the similarity metric. The enemy distance is the average distance within the case base to the validated case's three NUN's whereas the friend distance is the average distance to the validated case's three nearest like neighbours. A high ratio indicates a validated case that may harm the system's accuracy and would not be included in the case base. A conservative or aggressive approach to noise filtering can be applied by varying the ratio above which a validated case is not added to the case base. Noise filtering has only been used on known noisy datasets and has been applied using a conservative approach by not accepting validated cases with a ratio greater than 1.5.

Evaluation

In order to confirm that complexity-guided case discovery is useful we need to demonstrate that *useful* cases are discovered. Two complexity-guided algorithms have been compared with two benchmark algorithms to determine whether they result in case bases with increased accuracy. Five public domain classification datasets from the UCI ML repository (Blake, Keogh, & Merz 1998) have been used in the

evaluation. The selected datasets have varying numbers of features and classes, proportion of nominal to numeric attributes and level of noise. In addition, some datasets have missing values.

Complexity-guided case discovery cannot guarantee valid cases will be discovered. The objective is to supply a complete, candidate case to the case author to either accept or create a slight variation that corresponds to a valid case. However, this situation is difficult to replicate in an experimental evaluation because a domain expert is not available to validate the discovered cases. To simulate an expert our experimental design uses a pool of independent cases to act as an oracle. The candidate case then acts as a probe into this pool of cases to retrieve the most similar case from the oracle.

Each dataset was split into 5 independent folds with the folds being stratified to ensure a proportional representation of each class in each fold. One fold was used as the training set, one of the remaining four folds was used as the test set with the pool of cases being made up of the three unallocated folds. This process was repeated for each combination in turn resulting in 20 experiments (unique combinations of training set, test set and pool cases) for each dataset. There was no overlap between a training set and its associated test set and pool of cases.

The case base was initialised by randomly selecting a fixed number of cases from the training set. The starting size of the case base varied between 10 and 35 cases, depending on the dataset size and the difficulty of the problem. The algorithms were run on each trial on each dataset to discover between 5 and 40 cases in steps of 5. The results, averaged over the 20 runs, are plotted as a graph of the average accuracy for the test set for an increasing case base size, as an increasing number of cases are discovered. Test set accuracy is evaluated by a straightforward k -NN.

The experiments evaluate the effectiveness of the four algorithms described below on test set accuracy with a varying number of cases being discovered.

Algorithms

Four different case-discovery techniques have been implemented. Two are complexity-guided algorithms using a combination of the previously discussed techniques while the remaining algorithms provide benchmark comparisons.

All the algorithms identify two reference cases (a target case and its pair case) from within the case base. The main difference between the four algorithms is in their approach to identifying these reference cases.

- **COMPLEXITY** is our simpler complexity-guided algorithm. The complexity metric is calculated for each case and the 50% of cases with the highest complexity are ranked in descending order. Each case in turn (until the desired number of cases are discovered) is selected as the target case and its NUN is identified as its pair case. These two reference cases are used to create a candidate case to be between them by setting the candidate's feature values as either the mean or majority of the reference cases' feature values.

- **COMPLEXITY+** is a more informed algorithm that uses clustering to create a model of the case base. Figure 2 shows a simplified view of how this algorithm works in 2 dimensions. There are cases belonging to two classes with a class boundary between them. The cases are formed into clusters and the case with the highest complexity in each cluster is chosen as the target case (shown as a solid case). The target case's NUN is found (shown by an arrow) giving two reference cases and a candidate case is created to lie between them, as shown by the square.

The implementation of the algorithm involves the following stages. The complexity metric is calculated for each case. Clusters are formed and their complexity calculated as discussed earlier. The 75% of clusters with highest complexity are ranked in descending order of complexity. A target case is selected from each cluster in turn (until the required number of cases are discovered) and its NUN is selected as its pair case. A candidate case is created in the same manner as in **COMPLEXITY**. Where more cases are required than available clusters the stages are repeated including the complexity calculation and clustering. This incorporates the effect of the already discovered cases into the model.

- **COMPETENCE** uses competence-guided case discovery to create new cases between the nearest neighbour competence groups (McKenna & Smyth 2001). Two reference cases are selected from different competence groups that are nearest to each other. The candidate case's feature values are set using the feature values of the reference cases' related sets.

- **RANDOM** is an uninformed algorithm that selects two reference cases at random from the case base and then uses these reference cases to create a candidate case in the same way as **COMPLEXITY**. This process continues until the required number of cases have been discovered.

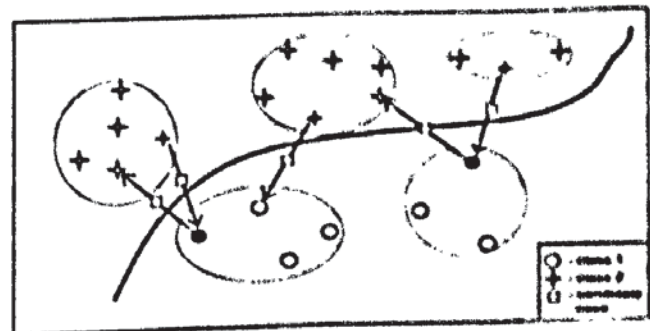


Figure 2: Illustration of **COMPLEXITY+**

Results

Significance is reported from a one-tailed paired t -test at 99% confidence, unless otherwise specified. Figure 3 (a) and (b) show average accuracy results for each case base size on the House Votes and Hepatitis domains. These are both binary classification problems with a bias to one of the classes. House votes is the larger data set with 435 cases

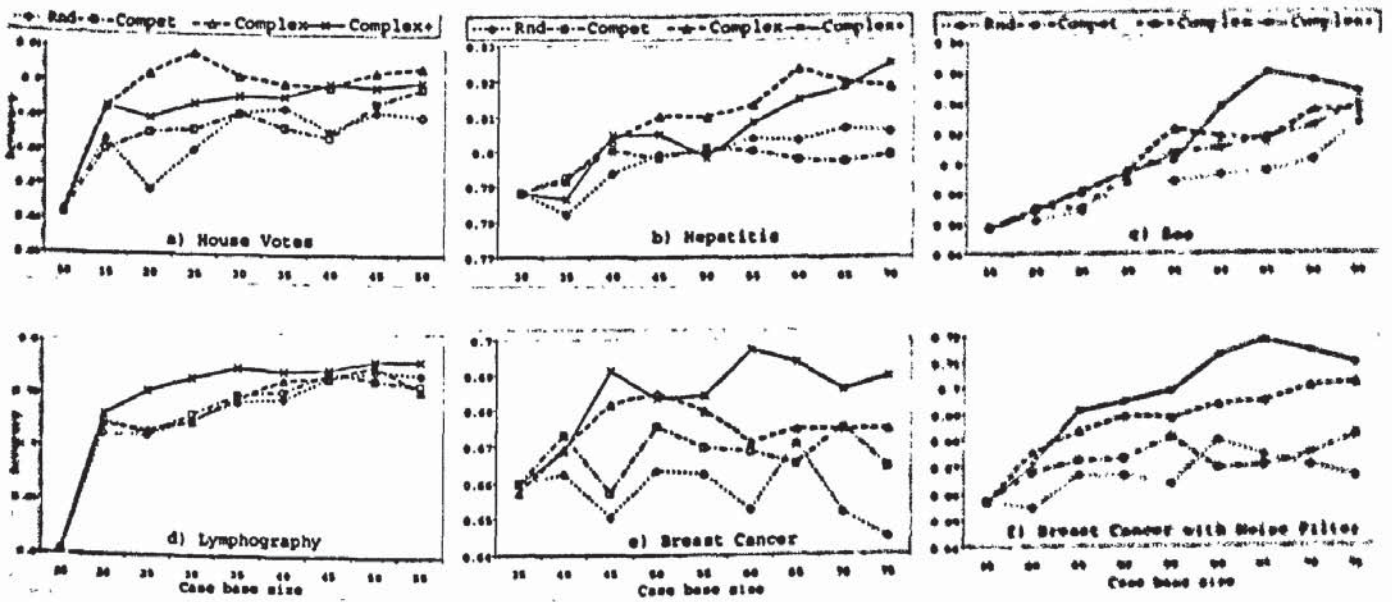


Figure 3: Accuracy of Growing Case Bases as Cases are Discovered

containing 17 nominal features while Hepatitis is a smaller data set of 155 cases represented by 20 mostly nominal features containing some missing values. As expected we see a significant improvement in accuracies on both House Votes and Hepatitis by the two complexity-guided algorithms (COMPLEXITY and COMPLEXITY+) over the RANDOM and COMPETENCE. Perhaps surprisingly, the simpler COMPLEXITY gives the best performance on these datasets. This might be explained by these being binary problems with high accuracy suggesting a more simple boundary than on the other datasets. The simpler algorithm, by concentrating on only few areas of the problem space, appears to perform well on this type of domain.

Average accuracy for the Zoo and Lymphography domains appear in Figure 3 (c) and (d). These are multi-class problems: Zoo has 101 cases split between 7 classes while Lymphography has 148 cases covering 4 classes. These domains have similar number of features (18 and 19) with no missing values. Zoo contains only nominal features whereas Lymphography contains both nominal and numeric. In both these domains COMPLEXITY+ produces the best performance with significant improvement over the other three algorithms. COMPLEXITY shows a significant improvement over RANDOM on the zoo domain but no difference over COMPETENCE. On Lymphography COMPLEXITY gave no improvement over either benchmark algorithm. The relatively poor performance of COMPLEXITY might be expected on these multi-class domains, as some of the classes contain a very small number of cases. In these situations COMPLEXITY will concentrate on providing cases to support the clusters with low representation and provide insufficient support to the rest of the problem space. In contrast, COMPLEXITY+ uses clustering to provide a more balanced distribution of new cases.

Figure 3 (e) shows average accuracy results on the Breast Cancer dataset. In Figure 3 (f) a noise filter, as described earlier, has been applied to all four algorithms for Breast Cancer. This is a binary classed domain with 9 multi-valued features containing missing data. The noise filter has been added because Breast Cancer is a more complex domain containing either noise or exceptional cases resulting in lower accuracies than the other domains. COMPLEXITY+ again produces the best performance with significant improvements over the other three algorithms. COMPLEXITY also shows a significant improvement over the two comparison algorithms in both experiments although the improvement over COMPETENCE without the noise filter is only significant at 95% confidence. The improved performance of COMPLEXITY+ over COMPLEXITY might again be explained by the simpler algorithm concentrating on supporting the noise or exceptional cases. It is interesting to see that, although the noise filter results in a small improvement in the performance of the benchmark algorithms it gives a large and significant improvement to the accuracies achieved by both the complexity-guided algorithms. This improvement is to be expected in noisy datasets because, by changing cases on class boundaries, the complexity-guided algorithms will have a greater tendency to pick noisy cases.

Evaluation Summary

The results from the significance tests, comparing the two complexity-guided case discovery algorithms with the benchmark algorithms on each dataset, are summarised in Table . The first two columns display the improvement with COMPLEXITY while the other two columns show significance results for COMPLEXITY+.

Overall COMPLEXITY+'s performance shows a significant improvement over the comparison algorithms on all the

Data Set	COMPLEXITY		COMPLEXITY+	
	vs. RANDOM	vs. COMPETENCE	vs. RANDOM	vs. COMPETENCE
House Votes	✓	✓	✓	✓
Hepatitis	✓	✓	✓	✓
Zoo	✓	no difference	✓	✓
Lymphography	no difference	no difference	✓	✓
Breast Cancer	✓	✓ (95%)	✓	✓
Breast Cancer-Noise	✓	✓	✓	✓

Table 1: Results summary according to significance

datasets and it provides the most consistent approach to case discovery of the algorithms studied. COMPLEXITY is shown to perform well on binary problems, particularly on simpler problems and on domains with low levels of noise, however, its performance on multi-class problems is only comparable with the benchmark algorithms.

The introduction of a noise filter stage gave significant accuracy improvements on the two complexity-guided discovery algorithms with Breast Cancer. This highlights the importance of noise filtering in noisy datasets.

Conclusions

The novel contribution of this paper is the use of a complexity metric and a case's NUN to guide the case discovery process by identifying interesting areas of the problem space. The idea of placing new cases on classification boundaries appears to be intuitively sensible in that it mirrors the approach of recently developed case base editing algorithms. COMPLEXITY and COMPLEXITY+, two new complexity-guided algorithms, were introduced and their effectiveness was demonstrated on 5 public domain datasets. In general, a significant improvement in test accuracy was observed with these new techniques compared to the random and competence-guided algorithms used as benchmarks. COMPLEXITY performed well on simple binary domains but suffered on multi class problems or on datasets containing noise. COMPLEXITY+, which incorporated a clustering stage, provided the most consistent performance across the range of datasets. A noise filter stage was found to enhance the performance of COMPLEXITY and COMPLEXITY+ on noisy datasets.

One limitation of the complexity-guided algorithms is that they restrict their search space to finding new cases within the problem space already covered by existing cases. Future work will focus on developing a complimentary approach for the very early growth stages of a case base, perhaps by using domain knowledge to seed the case base.

In this paper we have concentrated on providing support for the case author in the case discovery problem. However we are keen to see how the use of a complexity measure might be used more generally to provide support to the case author in other case base maintenance areas, such as case editing.

References

- Blake, C.; Keogh, E.; and Merz, C. 1998. UCI repository of machine learning databases.
- Brighton, H., and Mellish, C. 1999. On the consistency of information filters for lazy learning algorithms. In *Principles of Data Mining and Knowledge Discovery: 3rd European Conf.*, 283-288.
- Brighton, H., and Mellish, C. 2002. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery* 6(2):153-172.
- Delany, S. J., and Cunningham, P. 2004. An analysis of case-base editing in a spam filtering system. In *Proc. of the 7th European Conf. on Case-Based Reasoning*, 128-141.
- Liu, B.; Wang, K.; Mun, L.-P.; and Qi, X.-Z. 1998. Using decision tree induction for discovering holes in data. In *Proc. of the 5th Pacific Rim Int. Conf. on Artificial Intelligence*, 182-193.
- Liu, B.; Ku, L.-P.; and Hsu, W. 1997. Discovering interesting holes in data. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence*, 930-935.
- McKenna, E., and Smyth, B. 2001. Competence-guided case discovery. In *Proc. of the 21st Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence*, 97-108.
- McSherry, D. 2001. Intelligent case-authoring support in casemaker-2. *Computational Intelligence* 17(2):331-345.
- Smyth, B., and Keane, M. T. 1995. Remembering to forget. In *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence*, 377-382.
- Smyth, B., and McKenna, E. 2001. Competence models and the maintenance problem. *Computational Intelligence* 17(2):235-249.
- Wilson, D. R., and Martinez, T. R. 2000. Reduction techniques for instance-based learning algorithms. *Machine Learning* 38(3):257-286.
- Wilson, D. 1972. Asymptotic properties of nearest neighbour rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics* 2(3):408-421.
- Wiratunga, N.; Craw, S.; and Massie, S. 2003. Index driven selective sampling for CBR. In *Proc. of the 5th Int. Conf. on Case-Based Reasoning*, 637-651.

A Visualisation Tool to Explain Case-Base Reasoning Solutions for Tablet Formulation

Stewart Massie Susan Craw Nirmalie Wiratunga

School of Computing

The Robert Gordon University, Aberdeen

sm@comp.rgu.ac.uk

smc@comp.rgu.ac.uk

nw@comp.rgu.ac.uk

Abstract

Case Based Reasoning (CBR) systems solve new problems by reusing solutions of similar past problems. In complex tasks, such as configuration and design, it is not sufficient to merely retrieve and present similar past experiences. This is because the user requires an explanation of the solution in order to judge its validity and identify any deficiencies. Case retrieval with k -nearest neighbour relies heavily on the availability of cases, knowledge about important problem features and the similarity metric. However, much of this information, utilised by the system, is not transparent to the user. Consequently there is a need for tools that can help instil confidence in the system by providing useful explanations to the user. This paper proposes an approach that explains the CBR retrieval process by visualising implicit system design knowledge. This is achieved by visualising the immediate neighbourhood and by highlighting features that contribute to similarity and to differences. The approach is demonstrated on a pharmaceutical tablet formulation problem with a tool called FormuCaseViz. An expert evaluation provides evidence to support our approach.

1 Introduction

The problem being considered here is the formulation of a pharmaceutical tablet for a given dose of a new drug. Inert excipients (e.g. Lactose, Maize Starch, etc.) are chosen to mix with the new drug so that the tablet can be manufactured in a robust form. In addition to the drug, a tablet consists of five components each with a distinct role; i.e. Filler, Disintegrant, Lubricant, Surfactant, and Binder (see Figure 1). The formulation task entails identifying a suitable excipient and amount for each chosen component. Each chosen excipient must be suitable for its desired role and be compatible with each other and the drug. A more detailed description of the problem domain is available in [3].

A case-based reasoning (CBR) system solves new problems by reusing solutions from previously correctly solved similar problems [9]. Case retrieval is the first stage of the CBR cycle in Figure 2. For a tablet formulation, given a description of the new drug's chemical and physical properties together with specific requirements for the tablet, a similar case or a subset of similar cases useful for solving the new problem are retrieved from the case-base. Depending on the differences between the current problem and the retrieved cases some adaptation of the retrieved cases might be necessary before the retrieved solution can be reused. Here the proposed system solution

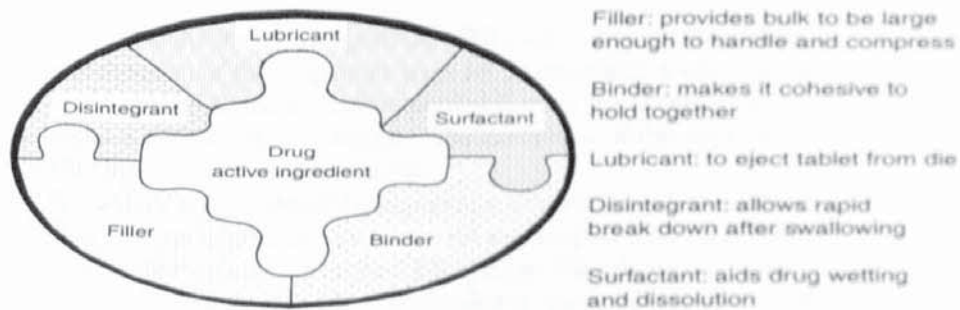


Figure 1: Tablet formulation problem

will be the excipients names and quantities that would enable the manufacture of a viable tablet. Subsequent stages include verification of the proposed solution and, if necessary retention of the new problem and the modified solution with the aim of reusing it in the future.

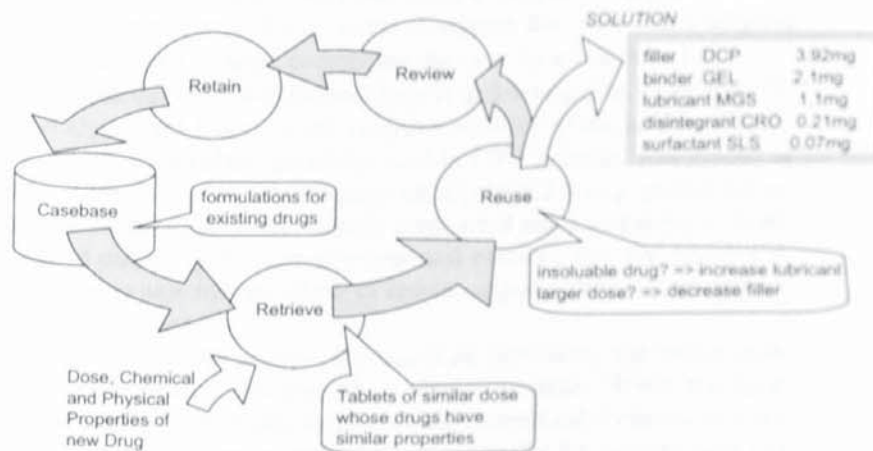


Figure 2: CBR cycle applied to the tablet formulation domain

An obvious advantage of CBR is its ability to present similar past experiences to an end-user instead of, for example a set of rules from a rule-base system. Clearly humans find it easier to relate to similar past experiences, however with domains such as tablet formulation where a case is described using 35 features simply presenting the cases will not aid the end-user's understanding of the reasoning behind the retrieval or even the proposed system solution. Mechanisms that aid human understanding of the system's problem solving process is vital because it helps instil confidence in the

system and may eventually determine the success of the system's deployment in the real world.

In this paper we look at the retrieval stage of the CBR cycle and tackle the problem of improving comprehension of this important stage by incorporating a visualisation tool. Our approach is applied to the tablet formulation domain but should be generally applicable across a wide range of domains. The usefulness of this approach is measured by conducting an expert user evaluation.

In Section 2 we explore in more detail the importance of explaining the solution in CBR. Section 3 discusses the information that different users expect from an explanation in order to increase their acceptance of the CBR system. The initial tablet formulation tool which provided the user with a textual solution is discussed in Section 4. A knowledge-light approach to providing an explanation of the solution is presented in Section 5. In Section 6 the design of the user evaluation is described along with the results. In Section 7 we review recent CBR research in this area. Finally we provide conclusions and recommendation for future work.

2 Solutions Require Explanations

Case Based Reasoning (CBR) is experience based problem-solving that mimics the approach often used by humans. One of the many advantages associated with CBR is its understandability as it can present previous cases to support or explain its conclusions [10]. Recent research provides evidence to support this view that a solution based on a previous experience is more convincing than one based on rules [4].

In contrast to the idea that the CBR methodology is understandable, King et al. [8] grade classification algorithms based on the *comprehensibility of the results*. The k -nearest neighbour (k -NN) algorithm, generally used in CBR systems, was graded at only 2 out of 5 by users, with only neural network algorithms being graded lower. One reason is that the similarity measure, usually compacted into a single value, hides the knowledge gained during system development and encoded into the design [2]. Revealing this knowledge aids interpretation of results, exposes deficiencies and increases confidence in the system.

Explanation of CBR solutions is typically based on presenting the single most *similar* case to the new problem, and possibly a similarity value. While this level of explanation might suffice in relatively simple, easily understood domains, it is not sufficient for tasks that are knowledge intensive. Individually the nearest case can provide an explanation. However, as with CBR solutions that can be improved by using several cases to provide a combined solution, likewise there is added value in providing an explanation based on several *similar* cases. This is particularly true if the similarities and differences within these cases can be made explicit; e.g. with the aid of visualisation.

In this paper we address this apparent contradiction that the CBR paradigm is transparent and understandable, yet the results of k -NN retrieval are not easy to comprehend. The CBR process draws heavily on knowledge held in the knowledge containers. Vocabulary, case and retrieval knowledge are three of the CBR *knowledge containers* proposed by Richter [13]. In order to encourage user acceptance of the

system the contents of these containers should be visible to the user. We present an approach that explicates this underlying knowledge to generate an explanation of the system solution formed on the basis of one or more nearest neighbours.

3 What Needs to be Explained

Knowledge intensive tasks require a better explanation than simply a proposed solution and a set of retrieved cases. This is particularly true of design problems, such as tablet formulation, where the case-base does not contain all possible designs. The proposed solution is only an initial draft, which may need to be adapted to compensate for differences between retrieved similar problems and the current problem at hand. The domain expert requires additional information and explanations to make the decision making process more transparent and to allow him to judge the validity of the solution. A visualisation tool for explanation must therefore highlight knowledge that not only drives the retrieval stage (e.g. retrieval knowledge containers), but also that suggests the need for adaptation (e.g. problem and solution differences).

3.1 Knowledge Containers

Knowledge elicitation for a CBR system, particularly applied to design tasks, can be substantial. A retrieve only CBR system will utilise the following knowledge containers:

- the case representation, generally containing two parts (the problem and solution), normally consists a set of attributes and values but can be a more complex structure. It is important for the user to be able to verify that this representation is suitable for the problem-solving task at hand.
- the case-base is the main knowledge source of a CBR system and usually determines its competence. It is not sufficient to expect the user to accept that the case-base provides representative problems. The user must be able to judge its quality and coverage in order to decide if it is suitable to address current problems. This will allow gaps in the case-base knowledge to be addressed and rectified.
- the retrieval process usually involves a similarity function that compares the cases held in the case-base with the new problem. This can be a Euclidean distance function or some domain specific function. The importance of individual features is often identified by feature weighting. The user needs to be able to decide if the similarity function is appropriate and if the importance of features is correctly represented.

This knowledge is often hidden from the user and can result in two effects: the user may accept the hidden knowledge as fact and not question it, or alternatively, confidence in the system may be harmed due to a lack of understanding of the hidden process. Either of these effects can be harmful to the usage of a CBR system.

3.2 The Proposed Solution

In addition to general information about the underlying CBR model being used, local information specific to the current problem must be visible to the user. This will allow a judgement to be made on the quality of the proposed solution and provide the relevant information to make manual adaptations. Visible, local information helps identify deficiencies in the current problem solving experience (e.g. quality of case-base, similarity function). It can be provided by comparing the new problem with either the case-base as a whole or with the most similar cases identified by the similarity function (its nearest neighbours). Local information is required in the following areas:

- **Similarities & Differences within Best Matching Cases and the Problem.** Easily interpretable information is required that allows the user to identify the attribute values that are common to both the problem and the best matching cases. More importantly it allows specific attribute value differences to be identified. This is the information needed to allow adaptation of the proposed solution. A one dimensional similarity value can hide these differences and is often not sufficient.
- **Relationship Between Neighbourhood & Case-base.** This allows the user to identify whether the case-base coverage is sufficient in the local region for this particular problem and allows an area of the problem space to be highlighted. Any deficiencies in coverage can be addressed by adding new relevant cases to the case-base.

4 Textual FORMUCASE

Our initial version of this application, called FORMUCASE, was developed along traditional CBR lines. Each case has a problem and solution represented by a list of attribute values. The problem attributes consist of five physical properties describing the drug itself and twenty chemical properties which describe how the drug reacts with possible excipients. All these attributes have numerical values. The solution has ten attributes; five with nominal values identifying the excipients used and five numeric values identifying the quantity of each excipient. When formulating a tablet for a new drug the attribute values representing the drug are entered and its nearest neighbours identified using the k -NN algorithm. The multi-component proposed solution to the problem is a weighted majority vote of its nearest neighbours to determine excipients and a weighted average for excipient quantities.

The output from FORMUCASE (see Figure 3) is presented in report format displaying the nearest neighbours, their problem and solution attribute values and their similarity to the new problem. The feature values of the proposed solution are then displayed. This retrieve-only system forms the first step in a tablet formulation. Differences between the new test problem and the retrieved cases may indicate the need to refine the predicted solution by manual adaptation.

An initial evaluation of FORMUCASE identified two problems. Firstly, confidence in the retrieval stage of the system is low as there was a reluctance to accept that

1 Nearest Neighbour: DrugT-200		Percentage match : 85.54%	
PROBLEM: Drug Solubility	: 0.8	SOLUTION: Filler; Amount	: Lactose 154.59mg
Drug Contact Angle	: 56.0	Disint; Amount	: Croscarmellose 9.9mg
Drug Yield Press	: 75.24	Binder; Amount	: PreGelStarch 6.9mg
Drug Yield PressFast	: 81.36	Lubricant; Amount	: MgStearate 3.43mg
Drug Dose	: 200	Surfactant; Amount	: null 0.0mg
Stabilities: 99.6;100; 100; 99.5; 0.0			
2 Nearest Neighbour: DrugQ-100		Percentage match : 70.27%	
PROBLEM: Drug Solubility	: 1.0	SOLUTION: Filler; Amount	: Lactose 162.2mg
Drug Contact Angle	: 42.0	Disint; Amount	: NaStarchGlyc 12.6mg
Drug Yield Press	: 24.84	Binder; Amount	: PreGelStarch 6.3mg
Drug Yield PressFast	: 45.6	Lubricant; Amount	: MgStearate 3.1mg
Drug Dose	: 100.0	Surfactant; Amount	: null 0.0mg
Stabilities: 100; 100; 100; 92.8; 0.0			
Suggested Tablet Formulation :			
Filler; Amount	: Lactose 167.04mg		
Disintegrant; Amount	: Croscarmellose 11.06mg		
Binder; Amount	: PreGelStarch 6.63mg		
Lubricant; Amount	: MgStearate 3.28mg		
Surfactant; Amount	: null 0.0mg		

Figure 3: FormuCase report format output

the similarity metric used provided similar cases to the current problem. Secondly, it was difficult to perform adaptation because differences between the new problem and the retrieved cases were not obvious. A revised version of this application, called FORMUCASEVIZ, was developed to alleviate these problems.

5 FORMUCASEVIZ

We demonstrate our approach to explanation using visualisation with this tablet formulation problem. Our hypothesis is that the visual version (FORMUCASEVIZ) will help explain the CBR process and increase user confidence in the solution. The problem and solution are displayed in parallel co-ordinate plots in order to address the issues discussed in Section 3.

A parallel co-ordinate graph's primary advantage over other types of statistical graphics is its ability to display a multi-dimensional vector or case in two dimensions. Figure 4 shows a plot with five dimensions. Each attribute is represented by a labelled vertical axis. The value of the attribute for each case is plotted along each axis. The points are then connected using horizontal line segments such that each case is represented as an unbroken series of line segments which intersect the vertical axes. Each axis is scaled to a different attribute. The result is a *signature* across n dimensions for each case. Cases with similar data values across all features will share similar signatures. Clusters of like cases can thus be discerned, and associations among features can also be visualised.

The basic layout of the graphical display for the tablet formulation task takes the form of three panels each containing a parallel co-ordinate graph (see Figure 5). The top graph contains twenty axes and provides attribute value information for the drugs chemical properties. The lower left graph contains five axes with the drugs physical

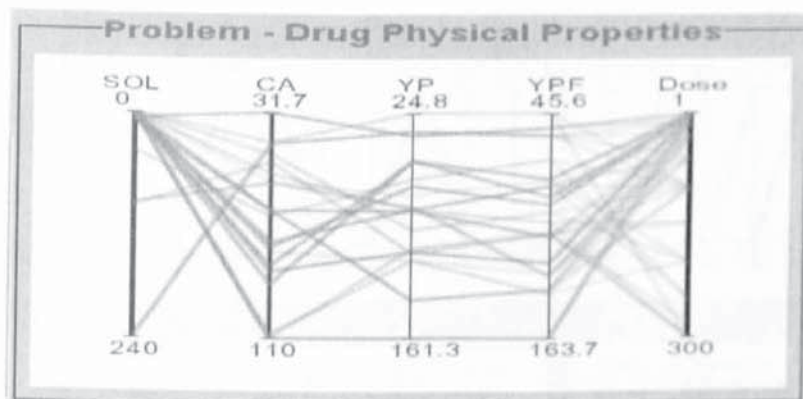


Figure 4: Parallel co-ordinate plot showing the drug physical properties of a case-base

properties and the lower right graph displays the solution attribute values. Thus the top and lower left panel contain attributes from the problem domain and the lower right graph contains attributes from the solution space.

Loading a case-base results in the vertical axes being drawn and labelled with an attribute's name and minimum and maximum value. The case lines, intersecting the axes, are also shown (see Figure 4). A visual picture of case-base coverage can now be seen with darker regions representing well covered areas of the problem space and gaps being visible as portions of the axis without case lines. The encoded retrieval knowledge in the form of feature weights is represented by the width of each axis. Figure 4 shows a case-base displayed on the drug physical properties graph. It can be seen that the attributes *SOL* and *Dose* have the highest weights.

We see in Figure 5 that on entering a new problem a black line representing it is drawn on the two problem domain graphs. This provides information on the local coverage provided by the case-base in relation to this particular problem. As no solution is yet available there is no black line representing the problem in the solution panel.

Figure 6 shows a solution to a new problem. The nearest neighbours are identified in the case-base and displayed as coloured dashed lines. The nearest neighbour solutions are also displayed in the solution panel along with the proposed solution for the new problem. A new axis is added to the drug physical properties problem panel showing the similarity value between the problem and its nearest neighbours along with labels for each case. This visualisation allows the similarities and differences to be viewed in terms of the real data aiding interpretation of the proposed solution and making the adaptation stage easier. For example, in Figure 6, it can be seen that the best matching cases disagree on which filler to use. *LAC* is the proposed solution but reference to the chemical stabilities show *DCP* would be a better choice as it has a higher chemical stability.

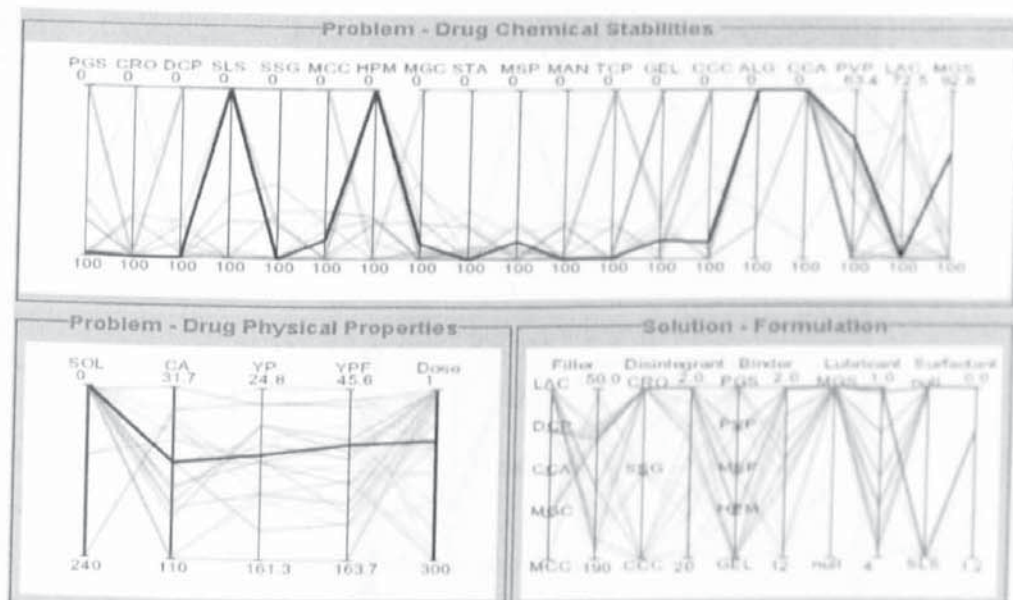


Figure 5: Output screen of FORMUCASEVIZ with an unsolved problem entered

5.1 Ordering the Attributes

The order or arrangement of the attributes is important when using parallel co-ordinate graphs. The arrangement can improve the visualisation by helping to identify trends or correlations within the case-base. Many approaches to multi-dimensional data visualisation arrange the attributes arbitrarily, possibly in the order that they appear in the case representation. We have taken the approach of arranging the attribute axis based on their similarity to each other in order to reduce line crossing on the graph. To achieve this axis arrangement we first use an axes similarity function to identify the pairwise similarities between the axes and then determine an arrangement so that similar axes are placed adjacent to each other.

An obvious way to measure axis similarity is to compare values across the cases. The similarity between axes A_i and A_j is measured using the attribute value similarity across the cases, rather than across the attributes as for case similarity. Thus, when case c_k is described by the n-tuple of attribute values (a_{1k}, \dots, a_{nk}) , the axis similarity from cases $c_1 \dots c_m$ is defined as follows:

$$Similarity(A_i, A_j) = \sum_{k=1}^m similarity(a_{ik}, a_{jk})$$

where similarity is the inverse Euclidean distance defined for individual (normalised) attribute values.

Determining a linear arrangement for the axes such that similar axes are placed close to each other is still not straightforward. We adopt the approach of first looking at the pairwise similarity values between the axes and picking the most similar pair.

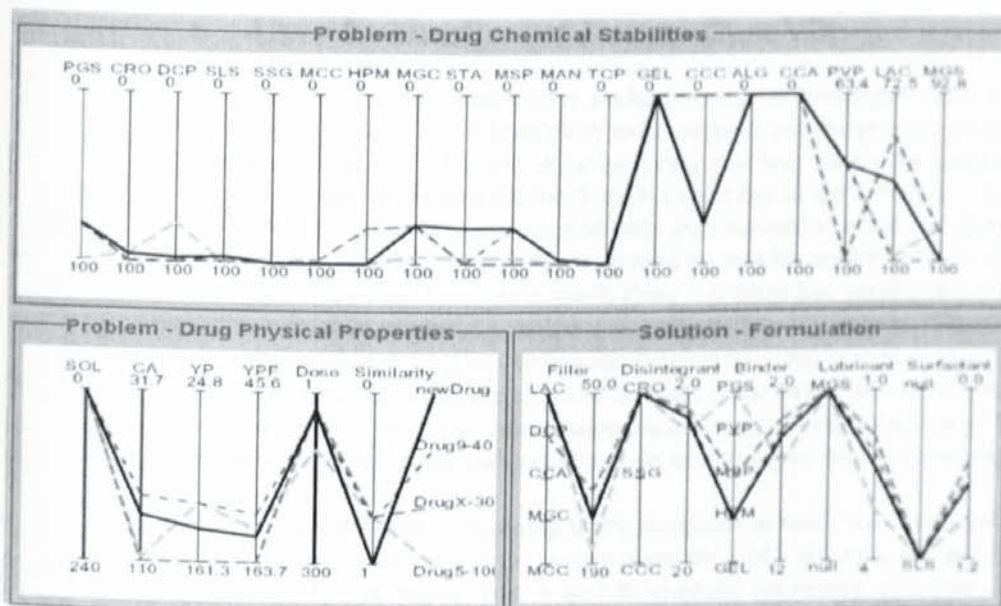


Figure 6: Output screen of FORMUCASEVIZ with a problem and proposed solution

These are placed to the left of the graph. The most similar unallocated axis is placed next to it. This process continues until all the axes have been allocated a position in the graph.

An alternative approach, which may give an optimal arrangement, is to find the order with the minimum total similarity when adjacent axis similarities are summed. However Ankerst et al [1] show that this problem is NP-complete. The use of a genetic algorithm or optimisation approach may be appropriate.

The arrangement of the axes can be carried out from a global or local context. The global arrangement looks at the whole case-base and takes no account of the current problem. This approach is best for looking at case-base coverage or when trying to identify trends within the case-base. It also has the benefit that it can be used prior to a problem being entered and is more stable as it remains unchanged as each new problem is entered. The local arrangement only looks at a portion of the case-base, typically around the new problem by only using its nearest neighbours in the calculation of the axes similarities.

FORMUCASEVIZ was implemented with the global arrangement on the two problem domain panels as it was found that the continual rearranging of axes gave problems in interpreting the results. However in other domains the advantages of a local approach may outweigh this disadvantage. No ordering of axes was applied to the solution panel as a fixed order was found to be more easily understood.

6 User Evaluation of FormuCaseViz

The purpose of the domain expert evaluation was to investigate how well FORMUCASE and FORMUCASEVIZ explain their solution and the process undertaken to arrive at the solution. This was done by looking at how easily the solutions could be interpreted and the confidence the domain expert has in the system's solution.

Three new tablet formulation problems were created to act as test queries for these evaluations. Test case 1 and 2 were created by making minor changes to an existing case held in the case-base. As a result, these test cases had similar nearest neighbours and a competent proposed solution was generated by the system. The third test case was created by removing an existing formulation from the case-base and making minor variations to it. This had the effect of creating a test case that had no similar cases in the case-base. There was considerable variation within the solutions of the retrieved nearest neighbour cases and confidence in the proposed solution was expected to be lower.

The questionnaire, containing thirty questions in total, was designed to ascertain the evaluators confidence in the system given the tool's ability to explain its reasoning. It contained three parts. Part A and B involved answering questions, as the three test cases were solved, using FORMUCASE and FORMUCASEVIZ respectively. The questions varied from specific questions e.g. *What is the similarity value of the best matching case?* to more abstract questions e.g. *Are you more confident in expicent prediction for the Filler or the Binder?*. Part C contained general questions comparing the advantages and disadvantages of two versions of the application.

Two domain experts were given both versions of FORMUCASE, a case-base and the three sample problems to solve. The evaluation required the expert to fill out the questionnaire while solving the three different test problems, on the same case-base, first with FORMUCASE and then with FORMUCASEVIZ. While the results of the evaluation are not published in detail here, we summarise our findings by highlighting some of the interesting observations.

- The experts agreed that FORMUCASEVIZ explains the CBR process of generating a solution better than the textual output version.
- There was a reluctance to accept a single similarity value alone as a measure of the case-base's competence to answer a specific new problem. In answer to the question *Does the case-base contain similar cases to the problem?* an unsure answer was usually given with FORMUCASE. In contrast, when presented with the same problem on FORMUCASEVIZ a definite and expected answer was always given.
- There was generally more confidence in the solutions provide by FORMUCASEVIZ and it was possible for alternative solutions to be suggested by the expert as would be required during the manual adaptation stage of the CBR cycle.
- The evaluators were better able to answer questions requiring them to identify differences within the nearest neighbours and between the problem and the neighbours. One evaluator commented *The graphical display is excellent and shows up similarities and differences in a very clear way.*

- Exact numerical values cannot be read from FORMUCASEVIZ as the values have to be interpolated from the axes. One expert highlighted this problem by commenting *the absence of easily readable numerical data is a big problem*. This deficiency needs to be addressed.

The positive results from our evaluation suggest that FORMUCASEVIZ provides a useful and more informative explanation of the proposed solution than FORMUCASE.

7 Related Work

CBR systems using decision tree guided retrieval typically provide their explanations by highlighting feature values of decision nodes traversed in order to reach the leaf node [4]. This is similar to the methods adopted in rule-based expert systems which often show rule activations [14]. Such rule-based explanation is not possible in systems using only k -NN retrieval because a set of discriminatory features is not identified as part of the algorithm. In these systems a typical approach is to present an explanation in terms of feature value differences between the query and the retrieved case. Cunningham et al. [4] suggest that explanations, expressed in terms of similarity only, can be useful in some domains (e.g. medical decision support) but is inadequate in others. McSherry's [12] approach to explaining solutions is based on identifying features in the target problem that support and oppose the predicted outcome. Discovery of the supporters and opposers of a predicted outcome is based on the conditional probabilities, computed from the cases available at run time, of the observed features in each outcome class.

Hotho et al. [6] provide explanations that are not solely similarity based. In their approach text documents are formed into clusters using a similarity metric and k -means clustering. The importance of the features or words in each cluster are ranked and the most *important* are used to represent the cluster. The relationship between clusters can then be identified using WordNet concept hierarchies. An explanation can now be given not only by the similarity of a document to other members in its cluster, but also on the relationship to other clusters.

Another approach to providing an explanation is through visualisation. McArdle & Wilson [11] present a dynamic visualisation of case-base usage by using a spring based algorithm. The algorithm uses the attraction and repulsion of the *springs* to spread the cases around a two dimensional graph in an attempt to preserve the n -dimensional distances between cases. This provides more insight into the similarity assessment than the usual single dimensional value. However, the knowledge held within the similarity metric is still hidden. Although this approach is used for supporting the maintenance of large case-bases it could also be adopted to visualise retrieved cases. An alternative approach is the parallel co-ordinate plot, originally proposed and implemented by Inselberg [7]. Falkman [5] uses this approach to develop an information visualisation tool, The Cube, which displays a case-base using three dimensional parallel co-ordinate plots. This approach allows the underlying data to be visualised as well as the similarity metric. We exploit this approach by also using a parallel co-ordinate plot to display the case-base but in addition we display underlying knowledge from the CBR knowledge containers and the retrieval process itself.

8 Conclusions and Future Work

A user gains confidence in a system that provides correct results. However confidence is also improved in systems where the decision making process is transparent and deficiencies can be identified and resolved. The explanation of results should be a key design criterion in CBR systems.

In this paper we have identified some of the reasons why CBR systems, particularly those using k -NN retrieval, are not as successful as they might be. We have presented an approach that can address some of these problems using a parallel co-ordinate visualisation of the problem and solution. This approach has been demonstrated on FORMUCASEVIZ in a tablet formulation problem domain in which thirty-five dimensional data is viewed in a single representation. A user evaluation confirmed that this explanation based approach made interpretation of the results easier than on the textual version, and better explained the CBR process. The need for exact numerical values to be available on the visualisation was also identified. While we have used tablet formulation in this paper our approach would be applicable across a wide range of CBR problem domains.

Future work will look at providing more local information related directly to the problem rather than to the case-base as a whole. This may involve a re-ordering of the attribute axes. Alternative, we may look at identifying or highlighting attributes in the problem domain that have the biggest impact in determining specific parts of the solution. This could be done by looking for correlations between axes in the problem and solution space. In addition we will also look at providing a more dynamic visualisation that allows the user to interact directly with the data, for example to change the problem or highlight certain areas of the case-base.

Acknowledgments

We acknowledge the assistance of PROFITS, Bradford University for funding the FORMUCASE demonstrator, providing the tablet formulation data and supplying willing domain experts for user evaluations.

References

- [1] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of IEEE Symposium on Information Visualization*, pages 52-60. IEEE Computer Society Press, 1998.
- [2] R. Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*. Springer, 2002.
- [3] S. M. Craw, N. Wiratunga, and R. Rowe. Case-based design for tablet formulation. In *Proceedings of the 4th European Workshop on Case-Based Reasoning*, pages 358-369. 1998.

- [4] P. Cunningham, D. Doyle, and J. Loughrey. An evaluation of the usefulness of case-based explanation. In *Proceedings of the 5th International Conference on Case-Based Reasoning*, pages 122–130. Springer, 2003.
- [5] G. Falkman. The use of a uniform declarative model in 3D visualisation for case-based reasoning. In *Proceedings of the 6th European Conference on Case-Based Reasoning*, pages 103–117. Springer, 2002.
- [6] A. Hotho, S. Staab, and G. Stumme. Explaining text clustering results using semantic structures. In *Principles of Data Mining and Knowledge Discovery. 7th European Conference*, pages 217–228. Springer, 2003.
- [7] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.
- [8] R. King, C. Feng, and A. Sutherland. Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9(3):259–287, 1995.
- [9] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA, 1993.
- [10] D. B. Leake. CBR in context: The present and future. In D. B. Leake, editor, *Case-Based Reasoning: Experiences, Lessons and Future Directions*, pages 3–30. MIT Press, 1996.
- [11] G. P. McArdle and D. C. Wilson. Visualising case-base usage. In *Workshop Proceedings of the 5th International Conference on Case-Based Reasoning*, pages 105–124. NTNU, 2003.
- [12] D. McSherry. Explanation in case-based reasoning: an evidential approach. In *Proceedings of the 8th UK Workshop on Case-Based Reasoning*, pages 47–55. 2003.
- [13] M. Richter. Introduction. In M. Lenz, B. Bartsch-Sporl, and S. Wess, editors, *Case-Based Reasoning Technology: From Foundations to Applications*, pages 1–15. Springer, 1998.
- [14] R. Southwick. Explaining reasoning: an overview of explanation in knowledge-based systems. *Knowledge Engineering Review*, 6:1–19, 1991.

Feature Selection and Generalisation for Retrieval of Textual Cases

Nirmalie Wiratunga¹, Ivan Koychev², and Stewart Massie¹

¹ School of Computing,

² Smart Web Technologies Centre,
The Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{nw|ik|sm}@comp.rgu.ac.uk

Abstract. Textual CBR systems solve problems by reusing experiences that are in textual form. Knowledge-rich comparison of textual cases remains an important challenge for these systems. However mapping text data into a structured case representation requires a significant knowledge engineering effort. In this paper we look at automated acquisition of the case indexing vocabulary as a two step process involving feature selection followed by feature generalisation. Boosted decision stumps are employed as a means to select features that are predictive and relatively orthogonal. Association rule induction is employed to capture feature co-occurrence patterns. Generalised features are constructed by applying these rules. Essentially, rules preserve implicit semantic relationships between features and applying them has the desired effect of bringing together cases that would have otherwise been overlooked during case retrieval. Experiments with four textual data sets show significant improvement in retrieval accuracy whenever generalised features are used. The results further suggest that boosted decision stumps with generalised features to be a promising combination.

1 Introduction

Past problem solving experiences captured in textual form present an interesting challenge to CBR system development. This is because experiences in unstructured form containing free text must first be mapped into structured cases before they can be meaningfully compared and reused for future problem solving. Textual CBR (TCBR) involves reuse of experiences that are in text form [14]. Unlike Information Retrieval approaches TCBR aims to develop case representation mechanisms that can better support knowledge-rich comparison of cases.

TCBR systems often access a variety of knowledge sources (e.g. domain specific thesauri, natural language parsers etc.) to establish an indexing vocabulary [5]. The general aim is to facilitate structured case representation and enhance retrieval. In this paper we investigate how introspective learning can be employed to automate the acquisition of the case indexing vocabulary [13]. We present techniques that are generally

applicable when textual experiences are pre-classified according to the types of problems they solve. Essentially we shall exploit implicit knowledge already existing in text documents to discover keywords that on their own or as a set in combination with others, are predictive of the problem class. The case indexing vocabulary will constitute just these selected keywords and so this process can be viewed as dimension reduction or feature selection.

Feature selection techniques employed by machine learning algorithms for supervised learning tasks such as classification are known to successfully improve accuracy, efficiency and comprehension of learned concepts [12]. Typically these techniques have been applied in problem domains consisting of structured cases. They have also been employed by CBR systems to identify relevant features for building an index for case retrieval [11]. A feature selection technique can be categorised as either being a filter or a wrapper approach. The wrapper approach uses feedback from the final learning algorithm to guide the search for the set of features. Generally this feedback ensures selection of a good set of features tailored for the learning algorithm but has the disadvantage of being time consuming because feedback involves learner accuracy ascertained from cross-validation runs. Filters are seen as data pre-processors and generally do not require feedback from the final learner. As a result they tend to be faster, scaling better to large datasets. Selection techniques presented in this paper fall under filter approaches which are particularly suited to processing of medium to large text collections.

In classification problems a *good* feature is one that is predictive of the problem class on its own or in combination with other features. Selection according to the performance of a combination of features is particularly useful for text data because there is often the need to identify similar meaning words that are used interchangeably (synonyms) and the same word being used with different meaning (polysemies). In both situations similar cases can be overlooked during retrieval if these semantic relationships are ignored. This paper introduces a novel feature selection technique that discovers and preserves semantic relationships in the case representation as part of the selection process. Boosted decision stumps are used for feature selection and semantic relationships are captured using association rule induction.

Section 2 describes the commonly used information gain based feature selection technique which is then used by the boosted feature selection technique in Section 3. The Apriori association rule learner is discussed in Section 4 and is employed as a means to capture semantic relationships between features. In Section 5, induced rules are utilised to form a generalised document representation and in doing so introduces novel ways of combining it with feature selection. Experimental results are reported on four textual classification tasks in Section 6. An overview of case representation and indexing issues in textual CBR research and how techniques presented in this paper relate to existing ones are discussed in Section 7, followed by conclusions in Section 8.

2 Feature Selection with Information Gain

We first introduce the notation used in this paper to assist presentation of the different feature selection techniques. Let \mathcal{D} be the set of all labelled documents, \mathcal{V} the set of all features which are essentially words. A document d is a pair (\mathcal{X}, y) , where \mathcal{X}

$= (x_1, \dots, x_{|\mathcal{W}|})$ is a binary valued feature vector corresponding to the presence or absence of words in \mathcal{W} ; and y is d 's class label [18]. The experiments in this paper use binary class domains so y is either 0 (negative class) or 1 (positive class). Let \mathcal{S} be the training subset containing labelled documents $\{d_1, \dots, d_n\}$.

The main aim of feature selection is to reduce $|\mathcal{W}|$ to a smaller feature subset size m by selecting features ranked according to some goodness criteria. The selected m features then form a new binary-valued feature vector \vec{x}' and a corresponding reduced word vocabulary set \mathcal{W}' , where $\mathcal{W}' \subset \mathcal{W}$ and $|\mathcal{W}'| \ll |\mathcal{W}|$. The new representation of document d with \mathcal{W}' is a pair (\vec{x}', y) .

A feature's discriminatory power is a useful gauge of its goodness and is commonly ascertained using the information gain (IG) score ([17], [16]).

$$IG(X, Y) = \sum_{x \in \{0,1\}} \sum_{y \in \{0,1\}} P(X = x, Y = y) \cdot \log_2 \frac{P(X = x, Y = y)}{P(X = x) \cdot P(Y = y)}$$

Here the probabilities are estimated from \mathcal{S} using m-estimates [15]. The information gain based ranking and selection of features is the base line algorithm used in this paper and we will refer to it as BASE (Figure 1).

```

m = feature subset size
BASE
  For each  $w_i \in \mathcal{W}$ 
    calculate IG score using  $\mathcal{S}$ 
  sort  $\mathcal{W}$  in decreasing order of IG scores
   $\mathcal{W}' = \{w_1, \dots, w_m\}$ 
  Return  $\mathcal{W}'$ 

```

Fig. 1. Feature selection with IG based ranking.

A feature goodness score like IG reflects a feature's ability to discriminate between classes. A possible shortfall with BASE is that selected features although having high scores may exercise their discriminatory power in similar ways. Consider documents from two mailing lists about computer hardware, one list containing messages about solving PC problems and the other dedicated to Apple Macs. An example of the top ranked words might be: "centris", "quadra", "eisa", "bus", "client", "server" etc. Here both "centris" and "quadra" are likely to suggest a hidden concept such as machine type. Similarly "eisa" and "bus" are likely to co-occur in similar documents and possibly relate to an implicit concept like internal architecture, while "client" and "server" are also features that can be viewed as belonging to a further implicit concept such as process communication. Ideally we would like to explicate these semantic relationships but firstly we need to ensure that as many of the hidden concepts are captured by at least a single representative discriminatory feature. This means that if we were restricted to select just three out of the six words a useful selection might be: "quadra", "eisa" and

"server" to cover each of the hidden concepts; instead of just the top three "centris", "quadra" and "eisa". What this example is highlighting is that selecting just the top ranked features with BASE can result in a feature set that is not particularly representative of hidden concepts thereby having a detrimental effect on case comparison. In the following section we combine IG based feature selection with boosting as a first step towards dealing with this problem.

3 Feature Selection with Boosted Decision Stumps

Boosting is known to improve the performance of learning algorithms particularly with tasks that exhibit varying degrees of difficulty [9]. The general idea of boosting is to iteratively generate several (weak) learners, with each learner biased by the training set error in the previous iteration or trial. Each learner works hard at solving training instances that were incorrectly classified in previous iterations. This is achieved by associating weights with instances in the training set and updating these weights at each trial. Weights of instances correctly solved by the most recent learner are decreased, and this has the effect of increasing weights of incorrectly classified instances. It means that at the next trial the learner is forced to work harder at solving these difficult instances. In order to classify a new test instance, the votes of each learner are combined to form a majority vote. Each vote is typically weighted by learner accuracy because it makes sense to trust those learners that have a higher accuracy on the training set.

An interesting approach to feature selection is to use boosting with a one-level decision tree, known as a decision stump, as the learning algorithm ([6], [8]). Constructing such a learner involves selecting a single feature, based on its ability to discriminate between classes [10]. For this purpose decision stumps are typically formed from features with high information gain. An example of two decision stumps from the binary classed computer hardware domain appear in Figure 2. Here a "+" denotes documents from the Apple mailing list and "-" the PC mailing list. With the "centris" stump the left leaf is formed by documents in which "centris" is present and the right leaf contains documents where it is absent. Predicting the class of a test document using this decision stump involves traversing the left or right branch leading to a leaf depending on the presence or absence of "centris" and labelling the document with the majority class at that leaf. Similar explanations hold for the stump having "bus" as the splitting feature. The stump error on the training set (err) is the percentage of the number of minority class documents in both branches.

Since a decision stump partitions the domain based on the values of a single feature, the set of stumps generated with boosting form the set of selected features. Therefore with m boosted iterations a set of m features are selected and these form the reduced feature subset \mathcal{W}' . The BOOST feature selection technique is shown in Figure 3. At each boosted iteration the feature with highest IG is selected forming the stump for the training set \mathcal{S} . Initially all n documents are assigned the same weight of $1/n$. With each trial these weights are updated so that the weights of correctly classified examples are reduced according to the error of the stumps. In practice once weights are updated, they need to be re-normalised so that their sum remains one. The impact of updated weights will be reflected in the IG scores where the prior and conditional probabilities are calcu-

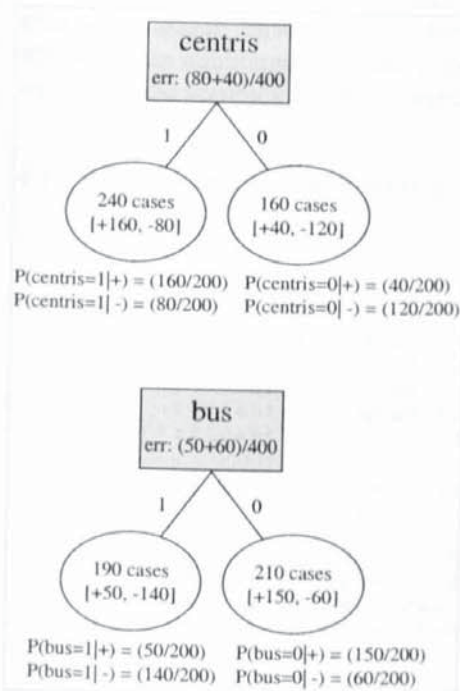


Fig. 2. Decision stumps.

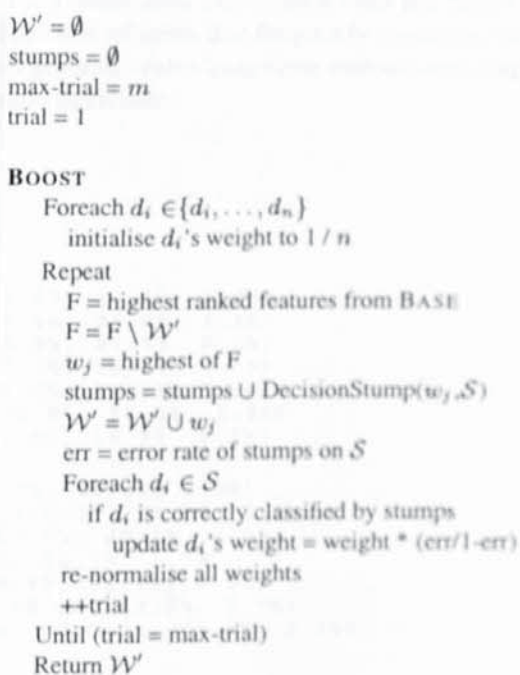


Fig. 3. Feature selection with boosted stumps.

lated on weighted documents, and this in turn will influence the feature selected in the next iteration when forming the stump. The boosting mechanism adopted here is similar to AdaBoost.M1 [9], the only difference being that updating of document weights is based on the error of the committee of stumps learned thus far, instead of the error of the most recent decision stump. With initial stumps containing features with higher IG scores the committee approach to updating document weights enables stumps from earlier iterations to exert a greater influence on feature selection.

Features that are discriminatory in similar ways have less opportunity to be selected with BOOST. However, with most tasks, information about which features co-occur with selected features can provide useful knowledge for case similarity, particularly in the presence of hidden concepts. In the next section we use an association rule learner to identify co-occurring features for selected features. A generalised feature space formed by applying these learned rules to selected features provides a richer case representation which in turn will enrich case comparison.

4 Feature Generalisation with Association Rule Induction

Apriori [1] is a well known association rule induction algorithm introduced for the market-basket analysis domain where one wishes to find regularities in people's shopping behaviour. It generates rules of the form $H \leftarrow B$, where the rule body B is a

conjunction of items, and the rule head H is a single item. Association rules are discovered in two stages. Firstly Apriori identifies sets of items that frequently co-occur, i.e. above a given minimum threshold. It then generates rules from these itemsets ensuring frequency and accuracy are above minimum thresholds.

4.1 Rule Generation and Selection

```
+r1:centri<- print (6.5%, 17.2%, 0.3%)
+r2:centri<- card (6.6%, 25.4%, 1.1%)
+r3:centri<- fpu (5.5%, 24.5%, 0.8%)
+r4:centri<- iisi (7.7%, 14.5%, 0.1%)
+r5:centri<- simm (9.0%, 16.3%, 0.3%)
+r6:centri<- quadra (10.8%, 24.0%, 1.5%)
+r7:centri<- lc (9.0%, 16.3%, 0.3%)

-r1:bus <- local (7.7%, 46.4%, 3.0%)
-r2:bus <- standard (10.3%, 31.5%, 1.2%)
-r3:bus <- window (13.6%, 29.5%, 1.2%)
-r4:bus <- id (20.5%, 28.3%, 1.7%)
-r5:bus <- drive (29.6%, 31.7%, 4.8%)
-r6:bus <- id local (9.0%, 42.0%, 2.7%)
-r7:bus <- drive local (10.3%, 37.0%, 2.1%)
```

Fig. 4. Example list of rules from the hardware domain.

An obvious analogy exists between frequently occurring itemsets in shopping transactions and frequently occurring words in a set of documents. This means that rules can be used to predict the presence of the head feature given that all the features in the body are present in the document. This means that a case satisfying the body even when the head feature is absent, will be considered closer to other cases that actually have the head feature present. Figure 4 lists two sets of rules generated for the hardware mailing list domain. The first rule set corresponds to rules generated with "centris" as the rule head and the other set with "bus" as the head. The class of documents from which these rules were induced are indicated by the rule prefix. This is important because co-occurrences of features are a signature of a particular class of documents.

In order to tie in a set of rules to a class it is necessary to constrain rule generation so that a rule's body contains features that are predictive of the same class as the rule's head, and learning is restricted to documents from this class. The predictive class of features is estimated according to class conditional probabilities. Going back to Figure 2, if "centris" is to be used as the head feature of the rule then the higher conditional probability, $P(\text{centris} = 1|+)$ indicates that it is most likely to appear in documents from the positive class. If instead "bus" is the head feature then the higher conditional probability $P(\text{bus} = 1|-)$ suggests the negative class.

An informed rule selection strategy is necessary because Apriori typically will generate many rules [3]. The percentages in Figure 4 are the coverage, accuracy and information gain for each generated rule. Generally the first two measures are used by

Apriori during rule generation to prune the search space. Here coverage (or frequency) is the percentage of documents in which a rule is applicable; and confidence (or accuracy) is the proportion of documents in which the rule prediction is correct. The third measures the gain in information due to the rule's body, and indicates how well the body is able to predict the presence or absence of the head feature. It is this measure that we have found most informative when selecting the K best rules from those generated. The three best rules predictive of each of the two head features (i.e. "centris", "bus") according to information gain are in bold.

4.2 Feature Generalisation

The objective of applying learned association rules is to improve case comparison by providing a more generalised case representation. Good generalisation will have the desired effect of bringing cases that are semantically related closer to each other that previously would have been incorrectly treated as being further apart. Association rules are able to capture implicit relationships (e.g. like synonyms) that exist between features. When these rules are applied they have the effect of squashing these features, which can be viewed as feature generalisation.

For a feature $w_i \in \mathcal{W}$, let \mathcal{R}_i be the set of association rules induced with w_i as the head feature, where $r_{ij} : w_i \leftarrow B_j$. Here the rule body B_j is a conjunction of features from $\mathcal{W} \setminus \{w_i\}$ and when true implies the presence of the head feature w_i . Given a document's initial representation $d = (\vec{x}, y)$ (i.e. using all features in \mathcal{W}), the generalised representation $d = (\vec{x}'', y)$ is obtained by applying $r_{ij} : x_i \leftarrow x_i \wedge \dots \wedge x_{ik}$, where $x_{ik} \neq x_i$, giving;

$$x''_i = \begin{cases} 1 & \text{if } x_i = 1 \\ 1 & \text{if } (\bigwedge_{k=1}^{n_j} x_{ik}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

All this means is that x''_i is instantiated with value 1 if either the head of the rule or its body is true, and is 0 otherwise. Consequently, the generalised new document representation \vec{x}'' tends to be less sparse than \vec{x} , because 0 values are likely to have their values flipped to 1. Essentially \vec{x}'' remains a binary valued feature vector, whose values indicate the presence or absence of a feature w'' , where $w'' \in \mathcal{W}''$, but $\mathcal{W}'' \not\subseteq \mathcal{W}$, since these features no longer correspond to presence or absence of single words.

Figure 5 illustrates how rules are used to generalise feature vectors. Here two forms of five trivial feature vectors are shown. The left table shows values for each vector using all the features in $\mathcal{W} = \{\text{"centri", "bus", "drive", "quadra", \dots}\}$, with the y column showing the document class. The right table shows the effect of generalisation after the sets of rules are applied. For sake of simplicity we use only the single best rule from each of the rule sets $\{\mathcal{R}_{centri}, \mathcal{R}_{bus}, \mathcal{R}_{drive}, \mathcal{R}_{quadra} \dots\}$; listed at the top of the figure. The first two rule sets contain a complete rule each: $\mathcal{R}_{centri} = \{+r6 : centri \leftarrow quadra\}$, $\mathcal{R}_{bus} = \{-r5 : bus \leftarrow drive\}$. So for example any rule from \mathcal{R}_{centri} (e.g. $+r6 : centri$) is applied to the left table's "centri" column on any document from the positive class, while rules from \mathcal{R}_{bus} are applied to the "bus"

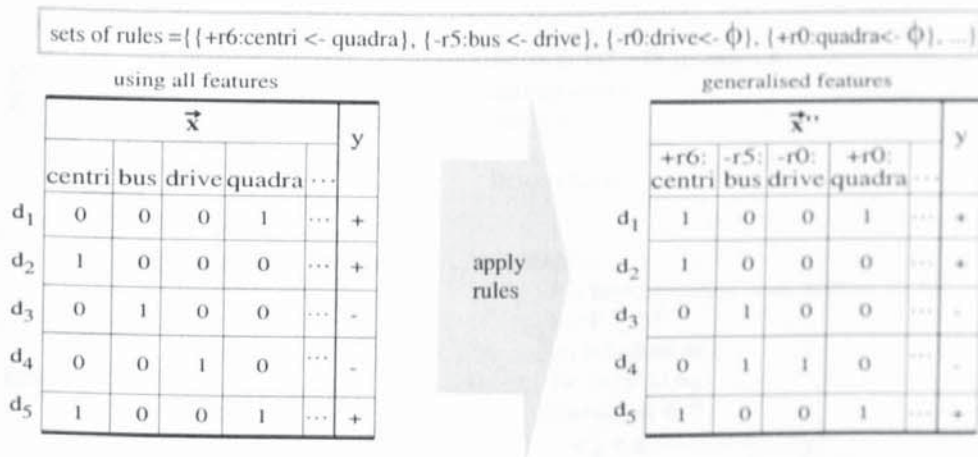


Fig. 5. Example of generalisation with rules.

column on any document from the negative class. The right table is the result of applying these rule sets. The other two rule sets: \mathcal{R}_{drive} and \mathcal{R}_{quadra} contain rules that have empty bodies. Such rules are not uncommon and indicate that Apriori was unable to find rules above specified minimum thresholds. Applying empty rules amounts to unchanged values, i.e. no generalisation takes place.

5 Combining Feature Selection with Generalisation

An obvious manner in which to perform generalisation is after feature selection. In Figure 6 BASEGEN does exactly this using BASE first to form \mathcal{W}' . It then uses \mathcal{W}' as a handle on ruleset generation, where a ruleset \mathcal{R}_i is generated for each selected feature $w'_i \in \mathcal{W}'$. This restricts the number of generated rule sets to m , so $|\mathcal{W}''| = |\mathcal{W}'|$. Here a rule $r_{ij} \in \mathcal{R}_i$ is of the form $r_{ij} : w'_i \leftarrow B_j$, where the rule body B_j is still a conjunction of features in $\mathcal{W} \setminus \{w'_i\}$, but the head now applies to a selected feature in \mathcal{W}' , where $\mathcal{W}' \subset \mathcal{W}$.

Interestingly we can also combine feature generalisation with boosted feature selection so that the boosted search for the best set of features is influenced at each iteration by the generalisation of the feature selected in the previous iteration. BOOSTGEN achieves this as shown in Figure 7. It calls *generalise* before forming the decision stump, as a result the decision stump is formed by splitting the training set according to the new generalised feature.

Generalisation after feature selection is attractive because generated rules will contain rule bodies that bring in features from the larger feature pool \mathcal{W} . In this manner both BASEGEN and BOOSTGEN are able to link selected features from \mathcal{W}' with other less frequently used features. This may be seen as supplementing selected features in \mathcal{W}' with background knowledge from \mathcal{W} . Additionally BOOSTGEN's boosted feature selection will tend to discover generalised features that are less likely to have overlapping semantic relationships with other generalised features.


```

 $\mathcal{W}'' = \emptyset; \mathcal{W}' = \emptyset$ 
BASEGEN
  call BASE to form  $\mathcal{W}'$ 
  Foreach  $d_i \in \mathcal{S}$ 
    Foreach  $w_j \in \{\mathcal{W}' \cap \mathcal{W}\}$ 
       $x''_{ij}$  = generalise( $x_{ij}, w_j$ )
       $w'_j$  = new generalised feature
       $\mathcal{W}'' = \mathcal{W}'' \cup w'_j$ 
  Return  $\mathcal{W}''$ 

```

```

generalise( $x, w$ )
   $\mathcal{R}$  = select-rules( $w$ )
  apply each rule in  $\mathcal{R}$ 
  generalising  $x$  to  $x''$ 
  Return  $x''$ 

```

```

select-rules( $w$ )
   $\mathcal{R}$  = rules with  $w$  as rule head
  sort  $\mathcal{R}$  decreasing order of rule IG
  break ties with coverage
  retain the best  $K$  in  $\mathcal{R}$ 
  Return  $\mathcal{R}$ 

```

```

 $\mathcal{W}'' = \emptyset; \mathcal{W}' = \emptyset; \text{stumps} = \emptyset$ 
max-trial =  $m$ 
trial = 1

```

```

BOOSTGEN
:
Repeat
  F = highest ranked features from BASE
  F = F \  $\mathcal{W}'$ 
   $w_j$  = highest of F
   $\mathcal{W}' = \mathcal{W}' \cup w_j$ 
  Foreach  $d_i \in \mathcal{S}$ 
     $x''_{ij}$  = generalise( $x_{ij}, w_j$ )
     $w'_j$  = new generalised feature
    stumps = stumps  $\cup$  DecisionStump( $w'_j, \mathcal{S}$ )
     $\mathcal{W}'' = \mathcal{W}'' \cup w'_j$ 
    err = error rate of stumps on  $\mathcal{S}$ 
  :
  ++trial
Until (trial = max-trial)
Return  $\mathcal{W}''$ 

```

Fig. 6. Generalisation after feature selection. Fig. 7. Generalisation with boosted selection.

6 Evaluation

Feature selection and generalisation techniques enable the mapping of textual documents into structured cases with which the case base is formed. Different case representations are formed using the 4 algorithms presented in this paper:

1. BASE, feature selection using the standard IG ranking (Figure 1);
2. BOOST, feature selection with boosting (Figure 3);
3. BASEGEN, generalisation after feature selection (Figure 6); and
4. BOOSTGEN, generalisation in combination with boosting (Figure 7)

The case retrieval performance using test set accuracy with 3 nearest neighbours is used to compare the above algorithms. A modified case similarity metric is used to refrain from treating the absence of words in the same way as the presence of words. This is because the presence of a word in documents is intuitively more important for measuring their similarity, than its absence. We accomplish this affect by weighting the similarity in non-present words by the inverse of the feature subset size. What this means is that as increasing number of features are used to represent documents, the influence of similarity due to the absence of similar words is reduced.

Textual cases were formed by pre-processing documents by firstly removing stop words (common words) and special characters such as quote marks, commas and full

stops (except for "!", "@", "%", "\$" because they have been found to be discriminative for some domains [17]). Remaining words are reduced to their stem using the Porter's algorithm. Essentially, \mathcal{W} is formed by all word-stems ($|\mathcal{W}| \approx 8000$) remaining after document pre-processing. For our experiments we use pre-processed documents from the following text corpora:

- LingSpam dataset has been formed to study the problem of spam. It contains 2893 email messages, of which 83% are non-spam messages related to linguistics, and rest are spam [17].
- 20 Newsgroups dataset is a corpus of about 20,000 Usenet news postings into 20 different newsgroups. One thousand messages from each of the twenty newsgroups were chosen at random and partitioned by newsgroup name [15]. For our experiments we use three sub-corpora, where the messages from two newsgroups are combined to form a binary classification as follows: Religion and Politics (RelPol); Apple Mac and PC Hardware (MacPc); and Space and Medical Science (SpcMed).

We created equal sized disjoint training and test sets, where each set contains 20% of documents randomly selected from the original corpus, preserving class distribution in the original corpus. For repeated trials, 15 such train test splits are formed. Significance is reported from a paired one tailed t-test with 99% confidence. The graphs show averaged accuracy on test set with increasing number of selected features.

6.1 Results

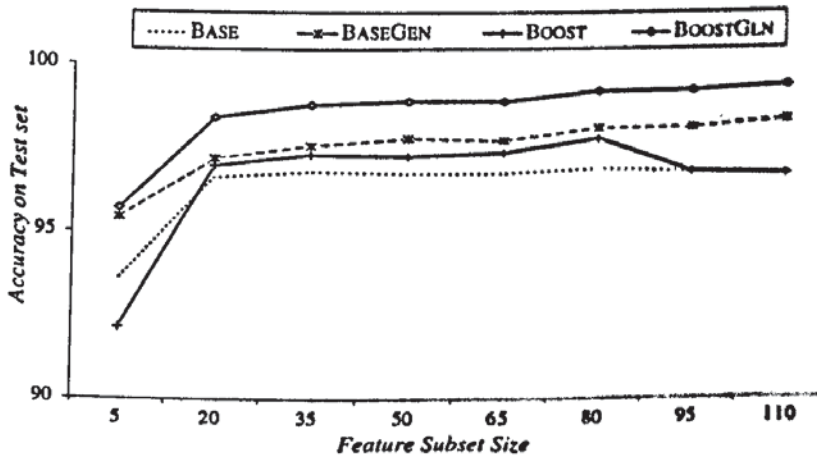


Fig. 8. Accuracy results for LingSpam.

The general behaviour of all four algorithms with the LingSpam corpus indicate an initial steep rise in accuracy (upto 20 features) after which there is hardly any improvement with increasing numbers of features (see Figure 8). The generalisation achieved

with BASEGEN has resulted in a small but significant increase in accuracy over BASE, while BOOST has only managed a slight improvement. However, BOOSTGEN's generalisation combined with boosting has significantly outperformed the other algorithms, achieving the highest accuracy approaching 99%. The overall accuracy results suggest that this domain is relatively easy because BASE achieves 93.6% accuracy with only five features and improves this accuracy to over 97% with twenty features and above. The reason for this is due to the nature of the LingSpam corpus, where there are a few very discriminatory features from non spam messages that are sufficient to differentiate spam messages.

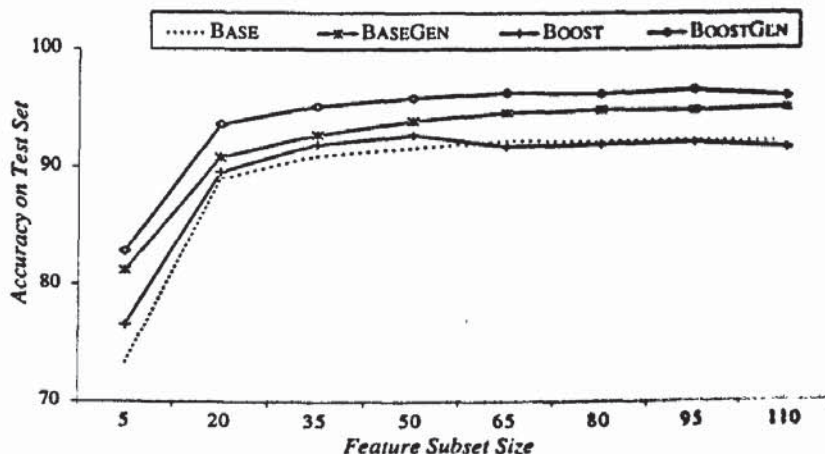


Fig. 9. Accuracy results for RelPol.

Figure 9 shows the results with the RelPol task. Compared to LingSpam the classification of documents into Religion and Politics seems to present a harder task because overall accuracy is lower. BOOST results are comparable to BASE where boosted feature selection shows improved accuracy with relatively smaller feature subset sizes. As before algorithms employing generalisation (BASEGEN and BOOSTGEN) outperform those without generalisation (BASE and BOOST), with BOOSTGEN having significantly improved performance over all other algorithms (including BASEGEN).

The results from the MacPc classification task appear in Figure 10. This task is expected to be the hardest, because similar terminology (e.g. monitor, hard drive) can be used in reference to both PC and Apple Mac hardware. Additionally the same hardware problem can be applicable in both mailing lists resulting in cross posting of the same message. Although boosting on its own has not improved accuracy, boosting combined with generalisation (BOOSTGEN) is significantly better than all other algorithms including BASEGEN at all feature subset sizes. Interestingly the accuracies for algorithms using generalisation (BASEGEN and BOOSTGEN) continue to rise with increasing feature subset sizes. The poor performance of BOOST can be explained by the relatively low discriminatory power of features in this domain. In fact selecting the most discriminatory feature followed by boosting of incorrectly classified documents can be harmful,

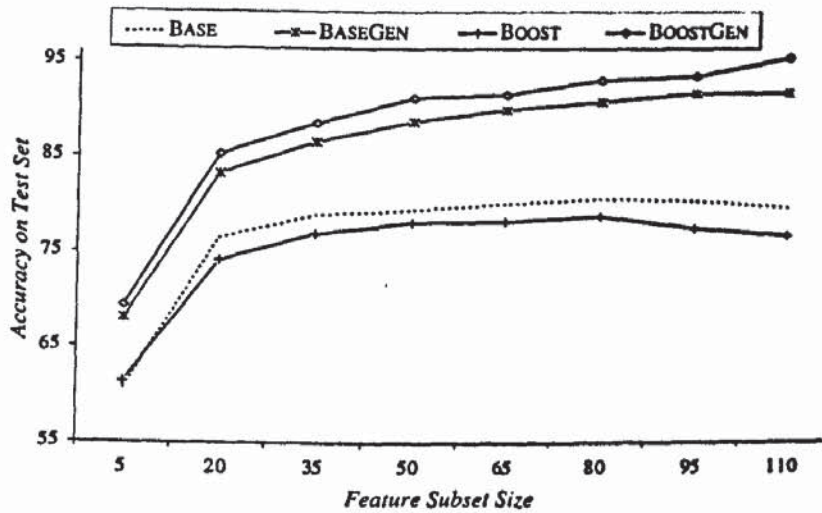


Fig. 10. Accuracy results for MacPc.

because updating of document weights prevents discovery of supportive features in subsequent boosted iterations.

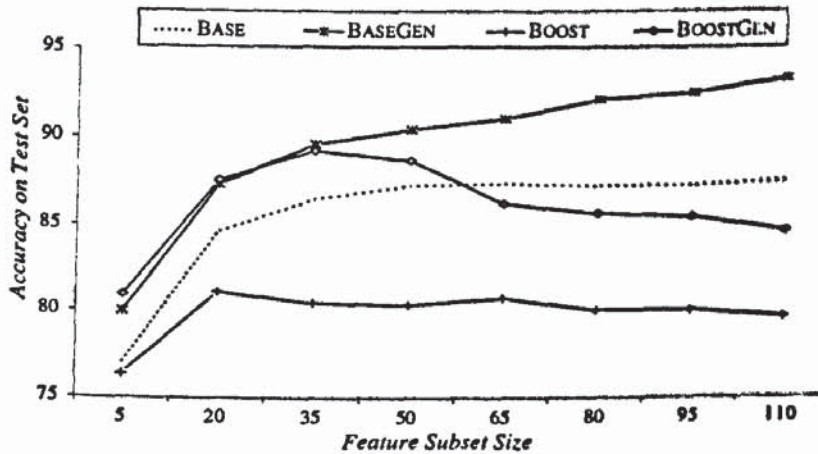


Fig. 11. Accuracy results for SpcMed.

A similar significant increase in classification accuracy with generalisation compared to without it is seen with the SpcMed domain (see Figure 11). Noticeably the overall winner here is BASEGEN having done significantly better than BOOSTGEN for the first time. Furthermore, boosting is not helpful and its performance is significantly worse than BASE. Closer examination of BOOST's results indicate over-fitting behaviour, because the accuracy on training set is higher than that of BASE's accuracy

on training set, but this gain is not reflected in test set accuracy. The generalisation used in BOOSTGEN maintains comparable performance to BASEGEN with up to 35 features, after which accuracy drops quickly as more features are used and over-fitting from boosting takes effect.

6.2 Evaluation Summary

Table 1. Results summary according to significance.

Data Set	Boosting		Generalisation	
	BOOST vs. BASE	BOOSTGEN vs. BASEGEN	BASEGEN vs. BASE	BOOSTGEN vs. BOOST
LingSpam	no diff.	✓	✓	✓
RelPol	no diff.	✓	✓	✓
MacPc	×	✓	✓	✓
SciMed	×	×	✓	✓

The results from the significance tests are summarised in Table 1. The first two columns convey the gain with boosting (BOOST vs. BASE and BOOSTGEN vs. BASEGEN); and the other two the gain with generalisation (BASEGEN vs. BASE and BOOSTGEN vs. BOOST). Overall feature generalisation improves algorithm performance significantly. It is worth noting that generalisation is able to continuously improve accuracy with increasing feature subset sizes with all domains, making it clearly more robust to over-fitting. Generally boosting is not helpful on its own, but BOOSTGEN combining boosting with generalisation achieves significant improvement over all other algorithms in 3 out of 4 domains.

7 Related Work

Current practice in TCBR system development show that the indexing vocabulary and similarity knowledge containers are typically acquired manually [19]. This is not surprising because of the ambiguous nature of free text. Although NLP tools can be applied to analyse free text they are often too brittle partly because they tend to analyse text from a purely linguistic point of view. Instead a piecemeal approach involving increasing levels of knowledge intensive containers have been identified as the basis for TCBR system development [13]. Generally these levels are broadly seen as connected with the case representation vocabulary or the similarity measure. Tools such as stemming, stop word removal and domain specific dictionaries form less intensive knowledge levels and are mostly automated. Acquiring semantic relationships between words typically form higher knowledge levels and are harder to automate and remain an important challenge.

The difficulty with acquiring an appropriate indexing vocabulary and the need for structured case representation within the law domain is discussed in [4]. The SMILE

system adopts a fine-grained sentence level class, whereby sentences are manually categorised into classes. It is interesting to note that although our approach does not explicitly assign classes at the sentence level, we also found it necessary to automatically link induced rules to applicable document classes. SMILE employs a decision tree based index scheme to partition the case base, but this is only possible after case sentences are manually marked-up (with words specified in a domain specific thesauri) to mitigate the synonym problem. We believe that our approach to feature generalisation with association rules helps automate the extraction of synonym relationships, provided that these relationships are already implicit in the textual case base.

Association rules have previously been used to reduce sparseness of initial user rating tables in collaborative recommendation [2]. Unlike traditional correlation based approaches Apriori is able to capture statistics about co-occurring features efficiently because it exploits the fact that no superset of an infrequent itemset can be frequent. Work presented in this paper combines feature selection with rule induction providing a useful strategy to manage rule generation and selection. Additionally the boosting in our approach attempts to capture features that tend to be orthogonal and with which hidden concepts can be discovered by exploiting rules generated by Apriori.

The aims of feature generalisation discussed in this paper are similar to those of Latent Semantic Indexing (LSI); a popular dimension reduction technique for text data. It uses singular value decomposition to map the word based feature vector representation into a lower dimensional latent space of artificial features [6]. Recently LSI was also integrated with textual case retrieval, where case similarity is computed on the basis of the lower dimensional case representation [7]. Unlike LSI our approach to feature vector generalisation explicitly captures hidden semantic relationships by way of association rules, enabling easier interpretation of generalised features during case comparison. Still it will be intriguing to see how the feature selection and generalisation techniques introduced in this paper compare with LSI based case representation.

8 Conclusions

The idea of feature generalisation and combining this with feature selection to form structured cases for textual retrieval is a novel contribution of this paper. Feature generalisation helps tone down ambiguities that exist in free text by capturing semantic relationships and incorporating these in the case representation. This enables a much better comparison of cases.

The two main approaches presented in this paper are feature selection with boosting and feature generalisation with association rules. Essentially feature selection helps with identifying discriminatory features while feature generalisation captures semantic relationships. Overall case representation with generalisation significantly improved accuracy over algorithms without generalisation, and promises great potential for automated acquisition of both the indexing vocabulary and the similarity containers. The effect of boosting is mixed where on its own gives modest improvement or even harmful in some domains, where it is more prone to over-fitting. Further research is needed to understand the relationship between types of problem domains and boosting per-

formance. However the best results in 3 of the 4 test domains were obtained by the combination of generalisation with boosting.

An interesting observation is that with feature selection and generalisation a more effective case retrieval is achieved even with a relatively small set of features. This is attractive because smaller vocabularies can effectively be used to build concise indices that are understandable and easier to interpret.

Acknowledgements

We thank Susan Craw, Rob Lothian and Dietrich Wettschereck for helpful discussions on this work.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.: Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, 307–327. AAAI/MIT Press (1995)
2. Alvarez, W., Ruiz, C.: Collaborative recommendation via adaptive association rule mining. In *Proceedings of the International Workshop on Web Mining for E-Commerce (2000)* 35–41
3. Borgelt, C., Kruse, R.: Induction of association rules: Apriori implementation. In *Proceedings of the 14th Conference on Computational Statistics (2002)*
4. Bruninghaus, S., Ashley, K.: Bootstrapping case base development with annotated case summaries. In *Proceedings of the Second International Conference on Case-Based Reasoning, ICCBR'99 (1999)* 59–73
5. Bruninghaus, S., Ashley, K.: The role of information extraction for textual CBR. In *Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR'01 (2001)* 74–89
6. Cai, L., Hofmann, T.: Text categorisation by boosting automatically extracted concepts. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2003)* 182–189
7. Chakraborti, S., Ambati, S., Balaraman, V., Khemani, D.: Integrating knowledge sources and acquiring vocabulary for textual CBR. In *Proceedings of the 8th UK-CBR workshop (2003)* 74–84
8. Das, S.: Filters, wrappers and a boosting based hybrid for feature selection. In *Proceedings of the 18th International Conference on Machine Learning*, Morgan Kaufmann (2001) 74–81
9. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (1996)*
10. Iba, W., Langley, P.: Induction of one-level decision trees. In *Proceedings of the Ninth International Workshop on Machine Learning (1992)* 233–240
11. Jarmulak, J., Craw, S., Rowe, R.: Genetic algorithms to optimise CBR retrieval. In Enrico Blanzieri and Luigi Portinale, editors, *Proceedings of the 5th European Workshop on Case-Based Reasoning*, Trento, Italy, Springer-Verlag, Berlin (2000) 137–149
12. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In *IJML94 (1994)* 121–129 Journal version in AIJ.
13. Lenz, M.: Defining knowledge layers for textual CBR. In *Proceedings of the 4th European Workshop on Case-Based Reasoning*, Dublin, Ireland, Springer Verlag (1998)
14. Lenz, M.: Knowledge sources for textual CBR applications. In *In Proceedings of the AAAI-98 Workshop on Textual Case-Based Reasoning*, Menlo Park, CA, AAAI Press (1998) 24–29

15. Mitchell, T.: *Machine Learning*. McGraw-Hill International (1997)
16. Pazzani, M. J., Muramatsu, J., Billsus, D.: Syskill and Webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR (1996) 54–61
17. Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C., Stamatopoulos, P.: A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6 (2003) 49–73
18. Salton, G., McGill, M. J.: *An introduction to modern information retrieval*. McGraw-Hill (1983)
19. Weber, R., Aha, D. W., Sandhu, N., Munoz-Avila, H.: A textual case-based reasoning framework for knowledge management applications. In *Proceedings of the 9th German Workshop on Case-Based Reasoning*. Shaker Verlag. (2001)

Visualisation of Case-Base Reasoning for Explanation

Stewart Massie, Susan Craw, and Nirmalie Wiratunga

School of Computing,
The Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{sm|smc|nw}@comp.rgu.ac.uk

Abstract. It is not sufficient for Case Based Reasoning systems to merely provide competent solutions. In complex tasks, such as configuration and design, the user requires an explanation of the solution in order to judge its validity and identify any deficiencies. Providing this explanation is not a straightforward task, particularly in systems using k -nearest neighbour retrieval, because much of the knowledge used to design the system is hidden from the user. This paper presents an approach to explaining the solution reached by a CBR system as well as highlighting differences between the target problem and most similar cases that may help to inform the adaptation process. This is achieved by presenting the best matching cases along with the system solution through a visualisation. The approach is demonstrated on a pharmaceutical tablet formulation problem with a tool called FormuCaseViz. An expert evaluation provides evidence of the potential benefits of our approach.

1 Introduction

Case Based Reasoning (CBR) is experience based problem-solving that mimics the approach often used by humans. One of the many advantages often associated with CBR is its understandability as it can present previous cases to support or explain its conclusions [8]. Recent research provides evidence to support this view that an explanation based on previous experience is more convincing than one based on rules [4].

In contrast to the idea that CBR techniques are understandable, King et al. [7] grade classification algorithms based on the *comprehensibility of the results*. The k -nearest neighbour (k -NN) algorithm, often used in CBR systems, was graded at only 2 out of 5 by users, with only neural network algorithms being graded lower. One reason is that the similarity measure, usually compacted into a single value, hides the knowledge gained during system development and encoded into the design [2]. Making this knowledge more accessible to users is an important challenge; we believe the potential benefits include simplifying the interpretation of results, exposing deficiencies in the reasoning process, and increasing user confidence in the system.

Explanation of CBR solutions is typically based on the single most *similar* case to the new problem, and possibly a similarity value. While this level of explanation might suffice in relatively simple, easily understood domains, it is not sufficient for tasks that are knowledge intensive. Individually the nearest case can provide an explanation. However, as with CBR solutions that can be improved by using several cases to provide

a combined solution, likewise there may be added value in providing an explanation based on several *similar* cases. We believe this is particularly true if the similarities and differences within these cases can be made explicit with the aid of visualisation.

In this paper we attempt to address the apparent contradiction that the CBR paradigm is transparent and understandable, yet the results of k -NN retrieval are not easy to comprehend. The user is expected to accept that the case-base contains representative problems and that the similarity measure used is appropriate to his problem. By hiding its similarity knowledge the system is not providing a satisfactory explanation of the solution and it is difficult for a user to have confidence in the system. We present an approach that makes this underlying knowledge and an explanation of the solution available. We demonstrate our approach on a tablet formulation problem domain. The usefulness of this approach is assessed in an expert user evaluation.

In Section 2 we review recent research on explanation in CBR. Section 3 discusses the information that different users need to explain a solution and to increase their acceptance of CBR systems. The problem domain on which we test our approach is discussed in Section 4. A knowledge-light approach to providing this information for a tablet formulation application is presented in Section 5. In Section 6 the design of the user evaluation is described along with the results obtained. Finally we provide conclusions and recommendation for future work in Section 7.

2 Related Work on Explanation in CBR

CBR systems using decision tree guided retrieval typically provide explanations by highlighting feature values of decision nodes traversed in order to reach the leaf node [4]. This is similar to the methods adopted in rule-based expert systems which often show rule activations [11]. Such rule-based explanation is not possible in systems using only k -NN retrieval because a set of discriminatory features is not identified as part of the algorithm. In these systems a typical approach is to present an explanation in terms of feature value differences between the query and the retrieved case. Cunningham et al. [4] suggest that explanations, expressed in terms of similarity only, can be useful in some domains (e.g. medical decision support) but is inadequate in others. McSherry's [10] approach to explaining solutions is based on identifying features in the target problem that support and oppose the predicted outcome. Discovery of the supporters and opposers of a predicted outcome is based on the conditional probabilities, computed from the cases available at run time, of the observed features in each outcome class.

Hotho et al. [6] provide explanations that are not solely similarity based. In their approach text documents are formed into clusters using a similarity metric and k -means clustering. The importance of the features or words in each cluster are ranked and the most *important* are used to represent the cluster. The relationship between clusters can then be identified using WordNet concept hierarchies. An explanation can now be given which is based not only by the similarity of a document to other members in its cluster, but also on the relationship to other clusters.

Another approach to providing an explanation is through visualisation. McArdle & Wilson [9] present a dynamic visualisation of case-base usage by using a spring based algorithm. The algorithm uses the attraction and repulsion of the *springs* to spread

the cases around a two dimensional graph in an attempt to preserve the n-dimensional distances between cases. This provides more insight into the similarity assessment than the usual single dimensional value. However, the knowledge held within the similarity metric is still hidden. Although this approach is used for supporting the maintenance of large case-bases it could also be adopted to visualise retrieved cases. An alternative approach is the parallel coordinate plot. Falkman [5] uses this approach to develop an information visualisation tool, The Cube, which displays a case-base using three dimensional parallel co-ordinate plots. This approach allows the underlying data to be visualised as well as the similarity metric. We exploit this approach.

3 What Needs to be Explained

Knowledge intensive tasks require a better explanation than simply a proposed solution and a set of retrieved cases. This is particularly true of design problems where the case-base does not contain all possible designs, and the proposed solution is only an initial draft, which may need to be adapted. The domain expert requires additional information and explanations to make the decision making process more transparent and to allow him to judge the validity of the solution. Further information is needed to explain both the CBR process and the proposed solution.

3.1 The CBR Process

Knowledge embedded in the CBR system in the form of stored cases and the similarity measure on which retrieval is based should be visible to the user:

- the case-base is the main knowledge source of a CBR system and usually determines its competence. The user must be able to judge its quality and coverage in order to decide if it is suitable to address current problems. This will allow gaps in the case-base knowledge to be addressed and rectified.
- the retrieval process usually involves a similarity function that compares the cases held in the case-base with the new query. This can be a Euclidean distance function or some domain specific function. The importance of individual features is often identified by feature weighting. The user needs to be able to decide if the similarity function is appropriate and if the importance of features is correctly represented.

This knowledge is often hidden from the user and can result in two effects: the user may accept the hidden knowledge as fact and not question it, or alternatively, confidence in the system may be reduced due to a lack of understanding of the hidden process. Either of these effects may have a negative impact on the acceptability of a CBR system.

3.2 The Proposed Solution

In addition to general information about the underlying CBR model being used, local information specific to the current query must be visible to the user. This will allow a judgement to be made on the quality of the proposed solution and provide the relevant information to make manual adaptations. Visible, local information helps identify

deficiencies in this particular problem solving experience (e.g. quality of case-base, similarity function). It can be provided by comparing the new query with either the case-base as a whole or with the most similar cases identified by the similarity function (its nearest neighbours). Local information is required in the following areas:

- **Coverage in the Neighbourhood of the Target Problem.** This allows the user to identify whether the case-base coverage is sufficient in the local region for this particular query and allows an area of the problem space to be highlighted. Any deficiencies in coverage can be addressed by adding new relevant cases to the case-base.
- **Similarities & Differences within Best Matching Cases and the Query.** Easily interpretable information is required that allows the user to identify the attribute values that are common to both the query and the best matching cases. More importantly it allows specific attribute value differences to be identified. This is the information needed for adaptation of the proposed solution. The overall similarity scores on which retrieval is based are inadequate for this purpose.

This additional information should be presented in an easily interpretable format that does not swamp the user with detail. We have employed a visualisation approach.

4 Problem Domain and FORMUCASE

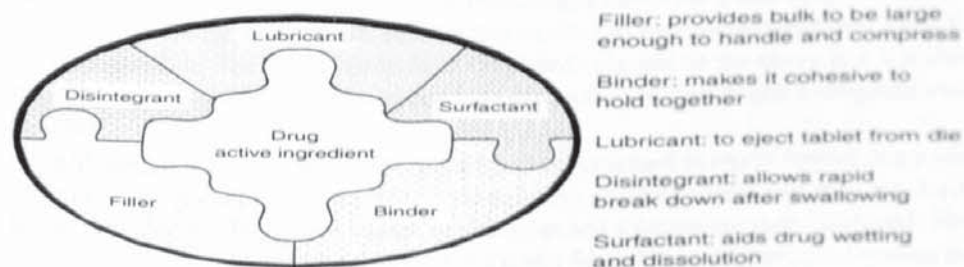


Fig. 1. Tablet formulation problem

FORMUCASE is a CBR system that formulates a tablet for a given dose of a new drug. This involves choosing inert excipients (e.g. Lactose, Maize Starch, etc.) to mix with the new drug so that the tablet can be manufactured in a robust form. In addition to the drug, a tablet consists of five components each with a distinct role; i.e. Filler, Disintegrant, Lubricant, Surfactant, and Binder (see Fig. 1). The formulation task entails identifying a suitable excipient and amount for each chosen component. Each chosen excipient must be suitable for its desired role and be compatible with each other and the drug. A more detailed description of the problem domain is available in [3].

1 Nearest Neighbour: DrugT-200		Percentage match : 85.54%	
PROBLEM: Drug Solubility	: 0.6	SOLUTION: Filler; Amount	: Lactose 154.59mg
Drug Contact Angle	: 55.0	Disint; Amount	: Croscarmellose 9.8mg
Drug Yield Press	: 75.24	Binder; Amount	: PreGelStarch 6.9mg
Drug Yield PressFast	: 81.36	Lubricant; Amount	: MgStearate 3.43mg
Drug Dose	: 200	Surfactant; Amount	: null 0.0mg
Stabilities: 99.6; 100; 100; 99.5; 0.0			
2 Nearest Neighbour: DrugQ-100		Percentage match : 70.27%	
PROBLEM: Drug Solubility	: 1.0	SOLUTION: Filler; Amount	: Lactose 162.2mg
Drug Contact Angle	: 42.0	Disint; Amount	: NaStarchGlyc 12.6mg
Drug Yield Press	: 24.84	Binder; Amount	: PreGelStarch 6.3mg
Drug Yield PressFast	: 45.6	Lubricant; Amount	: MgStearate 3.1mg
Drug Dose	: 100.0	Surfactant; Amount	: null 0.0mg
Stabilities: 100; 100; 100; 92.8; 0.0			
Suggested Tablet Formulation :			
Filler; Amount	: Lactose 167.04mg		
Disintegrant; Amount	: Croscarmellose 11.06mg		
Binder; Amount	: PreGelStarch 6.63mg		
Lubricant; Amount	: MgStearate 3.26mg		
Surfactant; Amount	: null 0.0mg		

Fig. 2. FormuCase output

Each case has a problem and solution represented by a list of attribute values. The problem attributes consist of five physical properties describing the drug itself and twenty chemical properties which describe how the drug reacts with possible excipients. All these attributes have numerical values. The solution has ten attributes; five with nominal values identifying the excipients used and five numeric values identifying the quantity of each excipient. When formulating a tablet for a new drug the attribute values representing the drug are entered and its nearest neighbours identified using the k -NN algorithm. The multi-component proposed solution to the query is a weighted majority vote of its k nearest neighbours to determine excipients and a weighted average for excipient quantities.

The output from FORMUCASE (see Fig. 2) is presented in report format displaying the nearest neighbours, their problem and solution attribute values and their similarity to the new query. The feature values of the proposed solution are then displayed. This retrieve-only system forms the first step in a tablet formulation. Differences between the new test problem and the retrieved cases may indicate the need to refine the predicted solution by manual adaptation.

5 FORMUCASEVIZ

We demonstrate our approach to explanation using visualisation with this tablet formulation problem. Our hypothesis is that the visual version (FORMUCASEVIZ) will help explain the CBR process and increase user confidence in the solution. The problem and solution are displayed in parallel coordinate plots in order to address the issues discussed in Section 3.

A parallel co-ordinate graph's primary advantage over other types of statistical graphs is its ability to display a multi-dimensional vector or case in two dimensions.

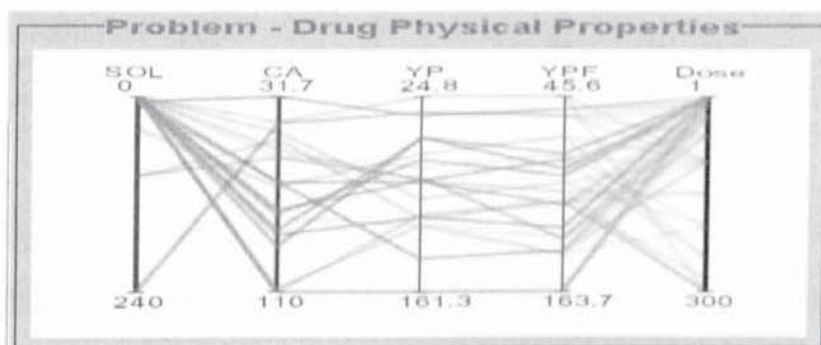


Fig. 3. Parallel co-ordinate plot showing the drug physical properties of a case-base

Fig. 3 shows a plot with five dimensions. Each attribute is represented by a labelled vertical axis. The value of the attribute for each case is plotted along each axis. The points are then connected using horizontal line segments such that each case is represented as an unbroken series of line segments which intersect the vertical axes. Each axis is scaled to a different attribute. The result is a *signature* across n dimensions for each case. Cases with similar data values across all features will share similar signatures. Clusters of like cases can thus be discerned, and associations among features can also be visualised.

The basic layout of the graphical display for the tablet formulation task takes the form of three panels each containing a parallel coordinate graph (see Fig. 4). The top graph contains twenty axes and provides attribute value information for the chemical stabilities of each drug with respect to the excipients commonly used in drug formulation. The lower left graph contains five axes with the drugs physical properties and the lower right graph displays the solution attribute values. Thus the top and lower left panel contain attributes from the problem domain and the lower right graph contains attributes from the solution space.

Loading a case-base results in the vertical axes being drawn and labelled with each attribute's name and minimum and maximum value. The case lines, intersecting the axes, are also shown (see Fig. 3). A visual picture of case-base coverage can now be seen with darker regions representing well covered areas of the problem space and gaps being visible as portions of the axis without case lines. The encoded retrieval knowledge, in the form of feature weights, is represented by the width of each axis. Fig. 3 shows a case-base displayed on the drug physical properties graph. It can be seen that the attributes *SOL* and *Dose* have the highest weights.

We see in Fig. 4 that on entering a new query a black line representing it is drawn on the two problem domain graphs. This provides information on the local coverage provided by the case-base in relation to this particular query. As no solution is yet available there is no black line representing the query in the solution panel.

Fig. 5 shows a solution to a query. The nearest neighbours are identified in the case-base and displayed as coloured dashed lines. The nearest neighbour solutions are also displayed in the solution panel along with the proposed solution for the new query. A

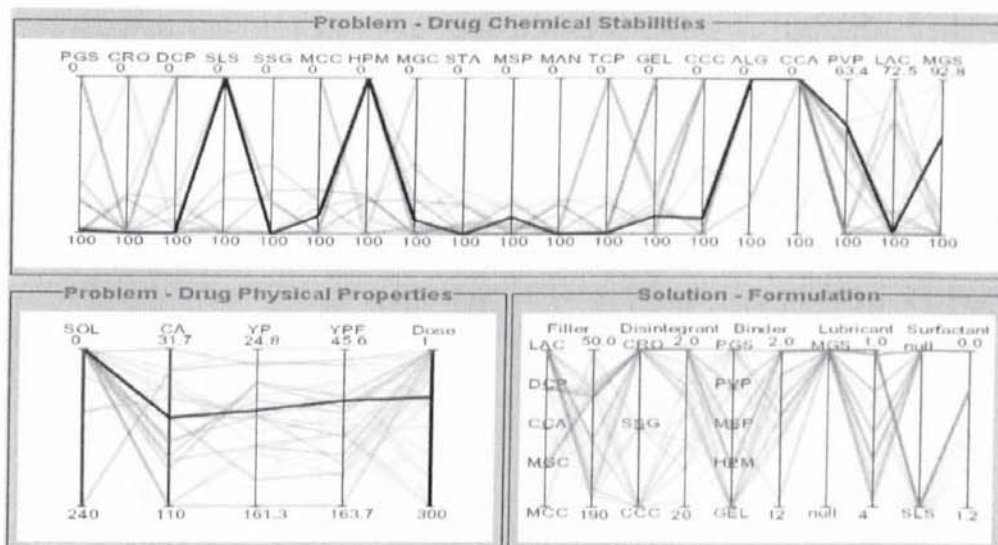


Fig. 4. Output screen of FORMUCASEVIZ with an unsolved problem entered

new axis is added to the drug physical properties problem panel showing the similarity of the query to each of its NN along with labels for each case. This visualisation allows the similarities and differences to be viewed in terms of the real data aiding interpretation of the proposed solution and making the adaptation stage easier. For example, in Fig. 5, it can be seen that the best matching cases disagree on which filler to use. LAC is the proposed solution but reference to the chemical stabilities show DCP would be a better choice for the new drug as it has a higher chemical stability.

5.1 Ordering the Attributes

The order or arrangement of the attributes is important when using parallel co-ordinate graphs. The arrangement can improve the visualisation by helping to identify trends or correlations within the case-base. Many approaches to multi-dimensional data visualisation arrange the attributes arbitrarily, possibly in the order that they appear in the case representation. We have taken the approach of arranging the attribute axes based on their similarity to each other in order to reduce line crossing on the graph. To achieve this axis arrangement we first use an axes similarity function to identify the pairwise similarities between the axes and then determine an arrangement so that similar axes are placed adjacent to each other.

An obvious way to measure axis similarity is to compare values across the cases. The similarity between axes A_i and A_j is measured using the attribute value similarity across the cases, rather than across the attributes as for case similarity. Thus, when case c_k is described by the n-tuple of attribute values (a_{1k}, \dots, a_{nk}) , the axis similarity from cases $c_1 \dots c_m$ is defined below where similarity is the inverse Euclidean distance defined for individual (normalised) attribute values.

6 Evaluating the Explanation

The purpose of the domain expert evaluation was to investigate how well FORMUCASE and FORMUCASEVIZ explain their solution and the process undertaken to arrive at the solution. This was done by looking at how easily the solutions could be interpreted and the confidence the domain expert has in the system's solution.

Two domain experts were given both versions of FORMUCASE, a case-base and three sample problems to solve. The evaluation required the expert to solve three different test problems on the same case-base. The evaluation is carried out first with FORMUCASE and then FORMUCASEVIZ. The experts were asked to fill out a questionnaire, containing thirty questions, that was designed to ascertain their confidence in the system given the tool's ability to explain its reasoning.

While the results of the evaluation cannot be presented in detail here, we summarise our findings by highlighting some of the interesting observations.

- The experts agreed that FORMUCASEVIZ explains the CBR process of generating a solution better than the textual output version.
- There was a reluctance to accept a similarity value alone as a measure of the case-base's competence to answer a specific query. In answer to the question *does the case-base contain similar cases to the query?* with FORMUCASE an *unsure* answer was usually given. In contrast, when presented with the same query on FORMUCASEVIZ a definite and expected answer was always given.
- There was generally more confidence in the solutions provide by FORMUCASEVIZ and it was possible for alternative solutions to be suggested by the expert.
- The evaluators were better able to answer questions requiring them to identify differences within the nearest neighbours and between the query and the neighbours. One evaluator commented *The graphical display is excellent and shows up similarities and differences in a very clear way.*
- Exact numerical values cannot be read from FORMUCASEVIZ as the values have to be interpolated from the axes. This is not ideal with one expert commenting *the absence of easily readable numerical data is a big problem.* This deficiency needs to be addressed.

The positive results from our evaluation suggest that FORMUCASEVIZ provides a useful and more informative explanation of the proposed solution than FORMUCASE.

7 Conclusions and Future Work

A user gains confidence in a system that provides correct results. However confidence is also improved in systems where the decision making process is understood and deficiencies can be identified and resolved. The explanation of results should be a key design criterion in CBR systems.

In this paper we have identified some of the reasons why CBR systems, particularly those using k -NN retrieval, are not as successful as they might be. We have presented an

approach that can address some of these problems using a parallel co-ordinate visualisation of the problem and solution. This approach has been demonstrated on FORMUCASEVIZ in a tablet formulation problem domain in which thirty-five dimensional data is viewed in a single representation. A user evaluation confirmed that this explanation based approach made interpretation of the results easier than the textual version, and better explained the CBR process. The need for exact numerical values to be available on the visualisation was also identified. While we have used tablet formulation in this paper our approach would be applicable across a wide range of CBR problem domains.

Future work will look at providing more local information related directly to the query rather than to the case-base as a whole either by re-ordering all the axes or highlighting specific axis where correlations can be identified. In addition we will look at providing a more dynamic visualisation that allows the user to interact directly with the data, for example to change the query or highlight certain areas of the case-base.

Acknowledgments

We acknowledge the assistance of PROFITS, Bradford University for funding the FORMUCASE demonstrator, providing the tablet formulation data and supplying willing domain experts for user evaluations.

References

1. Ankerst, M., Berchtold, S., Keim, D.A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of IEEE Symposium on Information Visualization*, IEEE Computer Society Press (1998) 52–60
2. Bergmann, R.: *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*. Springer (2002)
3. Craw, S. M., Wiratunga, N., Rowe, R.: Case-based design for tablet formulation. In *Proceedings of the 4th European Workshop on Case-Based Reasoning*, Springer (1998) 358–369
4. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In *Proceedings of the 5th International Conference on Case-Based Reasoning*, Springer (2003) 122–130
5. Falkman, G.: The use of a uniform declarative model in 3D visualisation for case-based reasoning. In *Proceedings of the 6th European Conference on Case-Based Reasoning*, Springer (2002) 103–117
6. Hotho, A., Staab, S., Stumme, G.: Explaining text clustering results using semantic structures. In *Principles of Data Mining and Knowledge Discovery 7th European Conference*, Springer (2003) 217–228
7. King, R., Feng, C., Sutherland, A.: Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence* 9-3 (1995) 259–287
8. Leake, D. B.: CBR in context: The present and future. In Leake, D. B. (ed.): *Case-Based Reasoning: Experiences, Lessons and Future Directions*. MIT Press (1996) 3–30
9. McArdle, G. P., Wilson, D. C.: Visualising case-base usage. In *Workshop Proceedings of the 5th International Conference on Case-Based Reasoning*, Springer (2003) 105–124
10. McSherry, D.: Explanation in case-based reasoning: an evidential approach. In *Proceedings of the 8th UK Workshop on Case-Based Reasoning*, (2003) 47–55
11. Southwick, R.: Explaining reasoning: an overview of explanation in knowledge-based systems. *Knowledge Engineering Review* 6 (1991) 1–19

What is CBR Competence

Stewart Massie Susan Crow Nirmalie Wiratunga
School of Computing
The Robert Gordon University, Aberdeen
sm@comp.rgu.ac.uk smc@comp.rgu.ac.uk nw@comp.rgu.ac.uk

Keywords: CBR, coverage, competence, case based reasoning

Case-based reasoning (CBR) is a popular approach to problem-solving because many of the knowledge engineering demands of conventional knowledge-based systems are removed. CBR solves new problems by re-using the solutions of previously solved similar problems. This research investigates the coverage, competence and problem-solving capacity of case knowledge with one of its aims being to develop a technique to model these aspects of a case-base and the clusters of cases within it.

What is competence?

Competence is a measure of how well a CBR system fulfils its goals. As CBR is a problem-solving methodology, competence is usually taken to be the proportion of problems faced that it can solve successfully. Other measures are also possible and indeed much recent research has looked at diversity as one of the possible goals of CBR [2]. Competence is a fundamental evaluation criteria vital to a system's performance. It is usually measured by either test set accuracy or cross-validation. These evaluation techniques are very time consuming to perform correctly, and methods of modelling the CBR system's competence could greatly reduce the need for this exhaustive evaluation. A good competence model would also help greatly in case-base maintenance research which has proposed numerous policies that either maintain or improve the competence of a system [4]. However there has been limited research aimed at modelling a case-base to provide values for competence in real situations.

Competence Models

Two different approaches have been used: the coverage approach [1] and the competence model approach [3]. Coverage assumes a finite problem space and attempts to measure the number of points within this problem space covered by the case-base. This empirical coverage only applies where the cases are represented by attribute vectors with nominal values providing a finite problem space, and simple adaptation methods identify all points in the problem space covered. This approach could not be applied to most CBR systems.

Smyth & McKenna's competence model assumes that the case-base contains a representative sample of problems. This is reasonable since a CBR system could not be a good problem solver if the case-base were not representative. Competence depends on properties like the number and density of cases. However, as competence is concerned with the range of target problems that a given system can solve, it also depends on the problem-solving ability of the system and must involve the retrieval and adaptation process of a system. The number and density of cases can be readily measured, but the problem of how to measure the problem-solving ability of a case in terms of its retrieval and adaptation characteristics is not so simple. Smyth and McKenna [3] suggest a four stage competence model which uses the representative assumption of the case-base to simulate the full domain. First leave-one-out testing is used to measure the problem-solving ability of a case using two important notions: coverage and reachability. Coverage of a case is the set of problems that case can solve; conversely, reachability is the set of all cases that can solve it. Next clusters of cases, called competence groups, are formed using their reachability and coverage sets to group cases that have overlapping sets. The coverage of each competence group is then measured by the group size and density. In the final step the overall competence of the case-base is simply the sum of the coverage of each group.

Experimentation

The true test of Smyth & McKenna's competence model is whether it reliably predicts the problem-solving ability of a CBR system. Experiments have been carried out using it applied to classification problems. In this scenario the reachability set of a case is its nearest neighbours with the same classification; i.e. the most similar k cases retrieved by the k Nearest Neighbour algorithm (k -NN) which have the same classification. Using this approach, the competence model was implemented on four classification datasets from the UCI Machine Learning Repository: iris, tic-tac-toe, zoo and house-votes.

Each of these datasets was split into training set and test set approximately in the ratio 70:30. Initially the training set was partitioned into fourteen disjoint sets. The smallest case-base was created using one of these sets, and a growing case-base was created by successively adding either one or two of these sets. Larger case-bases (800+ cases) were formed by adding duplicate cases to introduce redundancy into the case-base with 700 cases. The test sets were unseen target problems to calculate case-base accuracy and predict actual competence value for the case-bases.

The competence model was applied to each case-base and its predicted competence compared to the test set accuracy. Figure 1 shows the results for Tic-Tac-Toe and is typical of all four datasets. The left hand graph shows the accuracy obtained when using the unseen test set to evaluate the different sized training sets for 1-NN and 3-NN retrieval. The right hand graph shows the competence prediction made by the model for each different sized training sets for $k=1$ and $k=3$ in the leave-one-out testing stage of the model. It can be seen

from these graphs that there is little correlation between the model's predictions and test set accuracy. Hence the model does not produce an accurate prediction of the problem-solving ability of a CBR system in these situations.

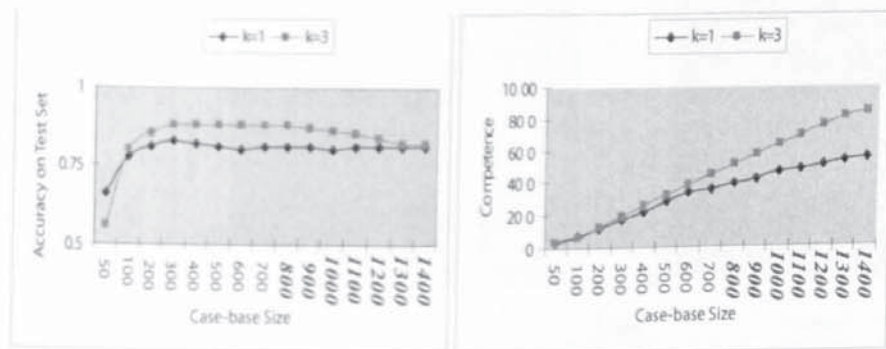


Figure 1: Graphs showing Accuracy on Test Set and Competence Prediction for different Case-base Sizes of the Tic-Tac-Toe Dataset

Proposed Competence Model

We wanted to investigate the low correlation between test set accuracy and the competence model prediction. A simple binary classification problem comprising two numerical features was created. This allows an easy visualisation of the problem space showing feature values, class and competence group membership for each case (Figure 2). Each case is represented by plotting a symbol on the graph according to the values of its two features. The two classes are distinguished by the shapes square and circle. The two competence groups are shaded differently in black or white.

Looking at a range of situations has identified two areas in which the model does not appear to reflect the problem-solving perspective. Figure 2 shows a visualisation of two case-bases for which the predicted competences for 3-NN is 15.6 in Figure 2a and 33.1 for Figure 2b. The same boundaries exist between the two classifications in each figure and so an identical problem is being viewed. However there is a far greater density of cases in Figure 2b. This increase in case density, including substantial redundancy within each competence group, has minimal effect on problem-solving ability but results in a large increase in predicted competence from the model. The average case density of a competence group does not appear to give a good measure of competence.

Similarly, the situation in which the boundaries between the two classifications are very different but the case-base composition is alike results in the predicted competence of the two situations being similar. However, the different boundaries between the classifications can result in one problem-solving situation being far more complex than the other. Hence the ability of the case-base for the more complex situation to correctly solve problems is less, i.e. its

competence is less. Again the model does not appear to adequately reflect the complexity of the problem being faced.

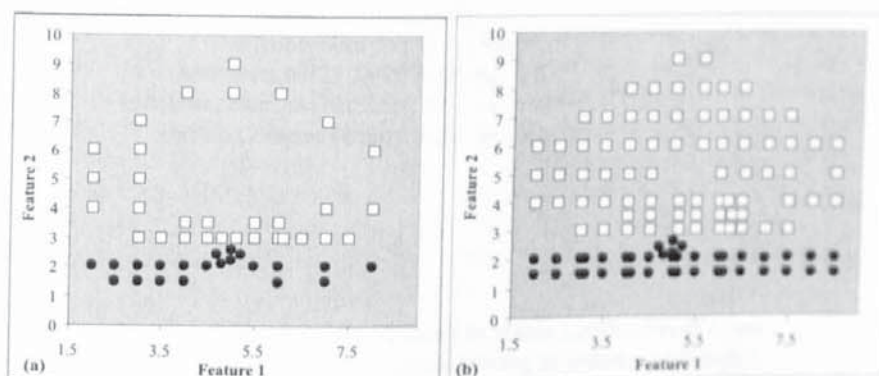


Figure 2: Visualisation of two different case-bases for a problem with the same boundaries between classification areas.

Conclusion

The research to date has highlighted the importance of modelling CBR competence in order to reduce the need for case-base evaluation experiments and to assist case-base maintenance research. Experiments have identified datasets in which the current models do not give a good correlation to the normal evaluation methods used to measure competence. Two reasons for this lack of correlation have been identified: group case density does not give a good measure of competence and problem complexity is not adequately reflected by the model. It is hoped that in future work a new model can be developed that concentrates on the boundary areas between classes.

References

- [1] D. McSherry. The case-recognition problem in intelligent case-authoring support. In *11th Irish Conference on Artificial Intelligence and Cognitive Science*, pages 180–189, 2000.
- [2] D. McSherry. Diversity-conscious retrieval. In S.Craw and A.Preece, editors, *Proceedings of the 6th European Conference on Case-Based Reasoning*, LNAI 2416, pages 219–233. Springer-Verlag, 2002.
- [3] B. Smyth and E. McKenna. Modelling the competence of case-bases. In *Proceedings of the 4th European Workshop on Case-Based Reasoning (EWCBR 98)*, pages 208–220, 1998.
- [4] D. Wilson and D. B. Leake. Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence*, 17(2):196–213, 2001.

Index Driven Selective Sampling for CBR

Nirmalie Wiratunga, Susan Craw, and Stewart Massie

School of Computing
The Robert Gordon University
Aberdeen AB25 1HG, Scotland, UK
{nw, smc, sm}@comp.rgu.ac.uk
<http://www.comp.rgu.ac.uk>

Abstract. In real environments it is often difficult to obtain a collection of cases for the case base that would cover all the problem solving situations. Although it is often somewhat easier to generate potential problem cases that cover the domain tasks, acquiring the solutions for the problems captured by the cases may demand valuable time of a busy expert. This paper investigates how a Case-Based Reasoning system can be empowered to actively select a small number of useful cases from a pool of problem cases, for which the expert can then provide the solution. Past cases that are complete, containing both the problem and solution, together with partial cases containing just the problem, are clustered by exploiting a decision tree index built over the complete cases. We introduce a Cluster Utility Score *CIUS* and Case Utility Score *CaUS*, which then guide case selection from these clusters. Experimental results for six public domain datasets show that selective sampling techniques employing *CIUS* and *CaUS* are able to select cases that significantly improve the accuracy of the case base. There is further evidence to show that the influence of complete and partial cases utilised by these scores needs also to consider the number of partitions created by the case base index.

1 Introduction

The main knowledge source in a Case-Based Reasoning (CBR) system is the case base. Typically a case consists of a problem description and the solution, or case label. When a new problem is encountered, a case with a similar problem part is retrieved from the case base, and its solution is reused. The availability of suitable cases is one of the arguments that supports the use of CBR for many problem-solving tasks. However, for some tasks the knowledge engineering effort can be significant [5].

There may be knowledge acquisition problems associated with other knowledge containers; e.g. specialised retrieval knowledge is required, or the problem solving relies on the availability of effective adaptation knowledge. However, it may also be caused by a lack of suitable cases. There may be few problem-solving experiences to record as cases, as in our tablet domain where the complex formulation task has been captured in relatively few manufactured tablets [4]. It could even be that the case base is biased with some areas of the problem space being very poorly represented. Although this is

often a consequence of a sparse case base, it can also occur in a plentiful case base, where there are holes in the coverage. Finally, a plentiful source of problems might be easily available but acquiring solutions for these problems might be harder: e.g. easy access to patient information related to a disease but acquiring laboratory results might be costly; or easy access to documents on the web but acquiring relevance feedback is time consuming.

The approach we investigate here is selective sampling, where although a source of new problems is readily available, the choice of cases for the case base is crucial because constraints limit the availability of case labels. This is a relatively common problem in a real environment, where labelling many problems with the expert's solution may require significant interaction with a busy expert. Unlabelled cases are often generated by analysing all labelled cases that are available with the aim of identifying holes in the domain [9], or random case generation might be adopted when there are no labelled cases initially. Unlike the labelling task, generating unlabelled cases does not typically require the assistance of an expert.

The work described in this paper performs an informed selection from a set of unlabelled cases. The expert must subsequently label only this subset, thereby reducing the demand on the expert. However, we must ensure that the informed selection of relevant cases does not hamper the competence of the case base by omitting cases that uniquely solve problems. Although we do not directly deal with the case discovery problem, we believe that useful insight in this direction can also be gained from the selective sampling approach presented in this paper.

The remainder of this paper describes our approach and evaluates it on several public domain case bases. Existing work in case selection and discovery are discussed in Section 2. A generic selective sampling process is presented in Section 3. It exploits a domain model created by labelled cases to sample unlabelled cases. Section 4 outlines how we use a case base index as our domain model to cluster all cases (labelled and unlabelled) in order to select unlabelled cases that are potentially useful, using heuristics described in Section 5. The approach is evaluated on several public domain datasets in Section 6, before we draw some conclusions in Section 7.

2 Related Work in Case Selection

The problem of unavailability of labelled cases and sample selection of relevant cases from a set of unlabeled cases falls under the paradigm of active learning and more specifically, selective sampling. Much work has been done in selective sampling of examples mainly related to training classifiers: using information about the statistical distribution of case feature values for nearest neighbour algorithms [8]; using a committee-based approach combined with expectation maximization for text classification [10]; and using a probabilistic classifier that selects cases based on class uncertainty for C4.5 [7]. Increasingly, estimation and prediction techniques with roots in statistics are being applied to classifiers with resulting improved accuracy [3].

Partitioning all labelled and unlabelled cases is a common approach that is employed by many active sampling techniques. Clustering cases in this manner helps identify interesting cases; i.e. those that have the potential to refine the domain model

learned thus far. But the use of cases for training classifiers differs from their use for a CBR system. In CBR, case retrieval is typically aided by a case base index, and retrieved cases may be directly reused to solve the new problem, or revised before being presented as a solution. The case base index which partitions cases into distinct problem solving areas in a CBR system will be exploited in this paper as a means to cluster labelled and unlabelled cases. Importantly by using the case base index we ensure that both the retrieval and adaptation stages influence the case selection process as opposed to simply exploiting the statistical distribution of cases.

Other CBR researchers utilise the CBR process when partitioning the cases. Smyth & Keane's coverage and reachability [13] are used to form competence groups of cases which can be used to solve each other [15]. These competence groups define the coverage of the case base, and allow narrow gaps to be identified where new cases can be proposed [11]. Competence groups are identified by applying the CBR retrieval and adaptation to cases in the case base. Boundary cases are pairs of cases, one from each of a pair of similar competence groups, chosen because they are most similar to each other. New cases are proposed that are midway between the boundary cases of the pairs of most similar competence groups. Their motivation is that close competence groups are more likely to merge with the addition of a new case that spans the narrow gap between them. This approach applies the model of the CBR reasoning to identify clusters of cases and hence small gaps between clusters.

CaseMaker [12] is an interactive knowledge acquisition tool that suggests potentially useful new cases for the case base by evaluating the coverage of possible new cases. It also applies the retrieval and adaptation knowledge to identify new cases.

This paper explores the same problem of identifying potentially useful cases to add to the case base. The case base index is used as a means to cluster the existing cases, to analyse the spread of existing cases and to suggest new cases that fill gaps identified by the clustering. Although retrieval and adaptation knowledge is not explicitly applied, the use of the index implicitly captures this knowledge. In contrast to competence-based case discovery [11], we select new cases that are dissimilar to existing ones, rather than discovering cases between close boundary cases. Although narrow gaps between large groups are interesting it is also vital to identify gaps within existing groups or even isolated gaps outwith existing groups.

3 Selective Sampling Process

The approach we investigate here is informed selection, where a source of new problems is available, and selective sampling identifies the most useful problems for which the expert should provide solutions, so that new cases can be added. This approach can also be used for case discovery where possible new problems are generated, and those that are selected for inclusion can be validated for consistency by the expert when he provides the solution.

Figure 1 outlines the selective sampling process for a set of unlabelled cases U by incorporating knowledge from labelled cases L . It would not be unusual to expect L to comprise a very small number of cases, while U would ideally contain a large set of cases. An initial model is created using the cases in the labelled set L . Using this

model, cases in both L and U are partitioned to form clusters. The aim is that each cluster contains cases that reflect the common problem solving behaviour abstracted by the model. In this paper we use a C4.5 decision tree as the model. Each cluster may contain zero or more cases from L , U , or both. The next step selects K clusters. This selection should ideally be guided by the labelled and unlabelled cases grouped together in a cluster. Once the K clusters are chosen, unlabelled cases are selected from these clusters. *Max-Batch-Size* is simply a constant that restricts the number of cases selected per sampling iteration.

```

L = set of labelled cases
U = set of unlabelled cases
LOOP
  model ← create-domain-model(L)
  clusters ← create-clusters(model, L, U)
  K-clusters ← select-clusters(K, clusters, L, U)
  FOR 1 to Max-Batch-Size
    case ← select-case(K-clusters, L, U)
    L ← L U get-label(case, oracle)
    U ← L \ case
  UNTIL stopping-criterion

```

Fig. 1. Selective sampling process

Case selection is incremental, and once labelled, by obtaining the solution from a domain expert or oracle, the new cases are appended to L , and U is updated accordingly, before a new domain model is created. So the aim then is to select cases that are most likely to trigger refinements (or improvements) to the domain model. This selection process iterates until a desirable level of accuracy is achieved on L , a sufficient number of new cases are added, or until the participation limit of the oracle is reached.

4 Case base Index : A Domain Model for Sampling

Several commercial CBR tools (e.g. RECALL, REMIND and KATE) use decision trees to index the case base in order to improve the efficiency of case retrieval [1]. Additionally these trees provide a useful means to explain the underlying reasoning for the retrieval. In previous work, we have shown that optimising the case base index improves the reuse stage [6] and that the partitions created by the case base index can further be exploited to acquire adaptation knowledge [16]. Therefore a CBR case base index forms a useful domain model since it identifies the areas of different problem solving behaviours in a case base. Here we look at how a case base index is created and how it can be used to form clusters.

4.1 Decision Tree Indexing

Figure 2 illustrates how a case base index can be created by inducing a C4.5 decision tree. The case base contains 20 (labelled) cases for a classification task with three classes X, Y and Z. The 3 decision nodes are tests associated with 3 of the features that describe the problem scenario captured by a case. Now let us see how the decision tree index is used within CBR retrieval. Assuming that a new problem is described as $f_1 = a$ and $f_2 = d$, then the tree would be traversed reaching the leftmost leaf node containing 5 labelled cases. These 5 cases form the relevant cases for the new problem, and by applying k -Nearest Neighbour (k -NN), we obtain the k nearest neighbours for the new case that lie within the leaf partition. Notice that here retrieval knowledge encompasses the nodes traversed and the feature weights that might be employed by k -NN. The new solution is obtained by reusing the majority solution suggested by the retrieved k cases, possibly with an adaptation stage added.

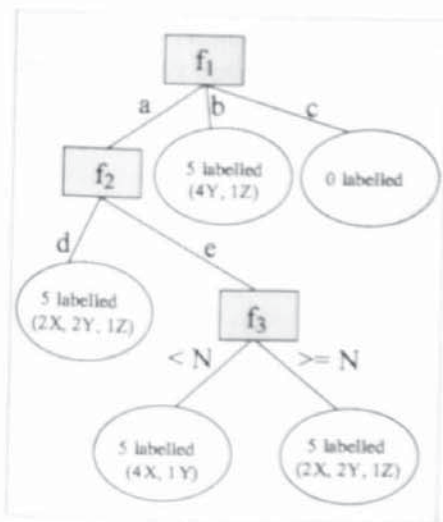


Fig. 2. Decision tree index

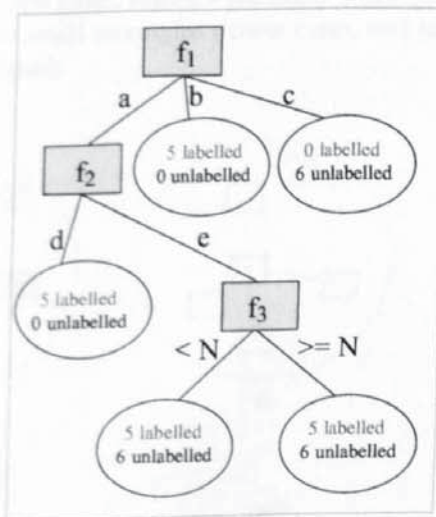


Fig. 3. Index for clustering

4.2 Index-Based Clustering

For sampling purposes we wish to partition all our cases so that cases in a given cluster share common retrieval and/or adaptation knowledge. The obvious candidates for a cluster are cases that are grouped together in a leaf node of the decision tree. Although leaf nodes of a case base index contain only labelled cases, the decision tree can be applied to unlabelled cases to allocate them to a leaf node, and hence a cluster. The problem description part of an unlabelled case alone is sufficient for tree traversal. Therefore, once an index is created using labelled cases, it is trivial to identify also the leaf nodes to which the unlabelled cases belong. Figure 3 shows the same index

4.1 Decision Tree Indexing

Figure 2 illustrates how a case base index can be created by inducing a C4.5 decision tree. The case base contains 20 (labelled) cases for a classification task with three classes X, Y and Z. The 3 decision nodes are tests associated with 3 of the features that describe the problem scenario captured by a case. Now let us see how the decision tree index is used within CBR retrieval. Assuming that a new problem is described as $f_1 = a$ and $f_2 = d$, then the tree would be traversed reaching the leftmost leaf node containing 5 labelled cases. These 5 cases form the relevant cases for the new problem, and by applying k -Nearest Neighbour (k -NN), we obtain the k nearest neighbours for the new case that lie within the leaf partition. Notice that here retrieval knowledge encompasses the nodes traversed and the feature weights that might be employed by k -NN. The new solution is obtained by reusing the majority solution suggested by the retrieved k cases, possibly with an adaptation stage added.

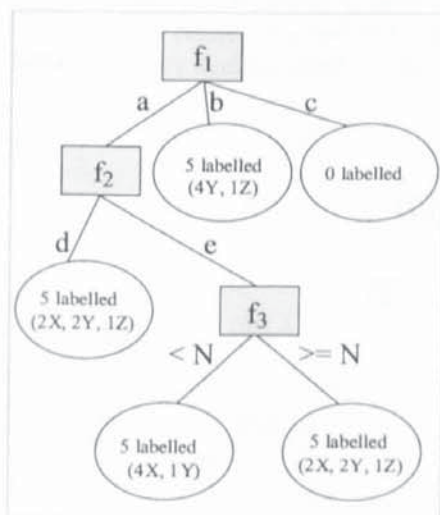


Fig. 2. Decision tree index

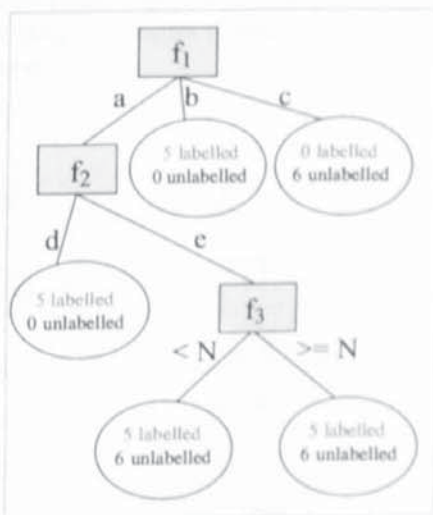


Fig. 3. Index for clustering

4.2 Index-Based Clustering

For sampling purposes we wish to partition all our cases so that cases in a given cluster share common retrieval and/or adaptation knowledge. The obvious candidates for a cluster are cases that are grouped together in a leaf node of the decision tree. Although leaf nodes of a case base index contain only labelled cases, the decision tree can be applied to unlabelled cases to allocate them to a leaf node, and hence a cluster. The problem description part of an unlabelled case alone is sufficient for tree traversal. Therefore, once an index is created using labelled cases, it is trivial to identify also the leaf nodes to which the unlabelled cases belong. Figure 3 shows the same index

after 18 unlabelled cases are introduced. Here we have 5 clusters: three containing either unlabelled or labelled cases and the other two containing a mixture of labelled and unlabelled cases.

Since the initial index was created with a small number of labelled cases, it is likely that the decision nodes, and hence the traversal paths, need to be refined. Therefore the clusters which are created according to the index are also bound to capture this incorrect traversal behaviour. The aim then is to identify clusters that contain useful cases in that they solve diverse problems whose solutions would provide useful new cases for the case base. Moreover the addition of these cases will bring about changes to the case base index, thereby refining the retrieval and adaptation stages.

5 Cluster and Case Selection

In this section we look at the sort of evidence that will aid the identification of interesting clusters that are likely to contain useful new cases. Figure 4 illustrates a detailed view of the 5 clusters formed in Figure 3. The small rectangles denote cases, and labelled cases are distinguished by shading and labels.

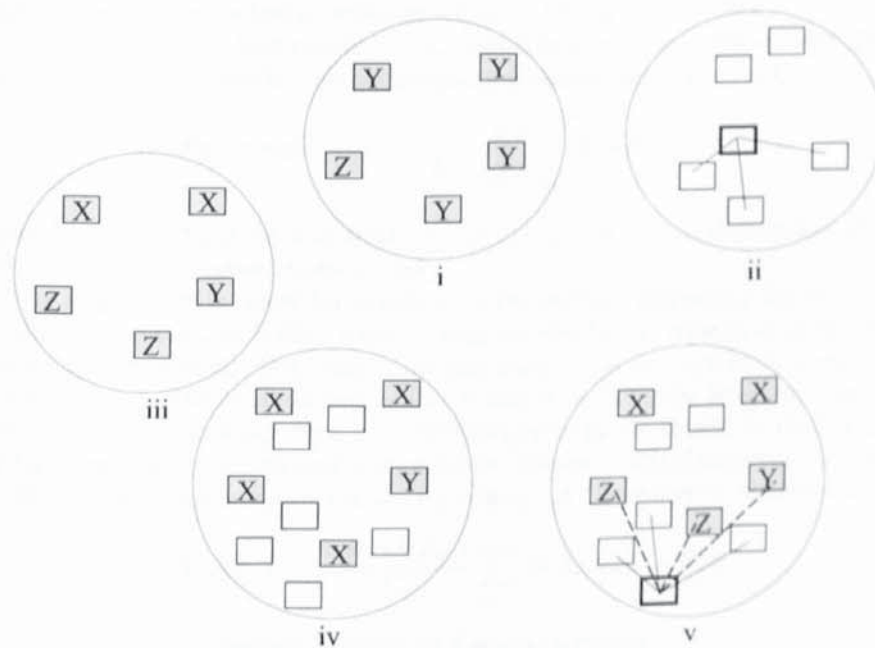


Fig. 4. Selecting a cluster

Since clusters (i) and (iii) do not contain any unlabelled cases there is no need to concentrate on them, because they are not a source of new cases. In contrast cluster (ii) contains 6 unlabelled cases and, since there are no labelled cases representing this area

of problem solving, it would be useful to select this cluster. If however we had to select between clusters (iv) and (v), each containing some labelled and unlabelled cases, we would wish to pick the one that contains atypical cases; i.e. the cluster with low intra-cluster similarity. Additionally, the fact that there is a greater mixture of labelled cases with cluster (v) indicates more uncertainty, and so it should be considered most fruitful.

Once an interesting cluster is identified what criteria should we employ to ascertain the usefulness of a new case? Certainly a case that is representative of other unlabelled cases is useful because by selecting such a case we cover a greater number of problem solving situations. In contrast selected cases also need to be different from labelled cases that are already in the cluster. With cluster (ii) the bold rectangle indicates a candidate case that is sufficiently representative of (ii), in that it has the least distance to its unlabelled 3 nearest neighbours. With cluster (v) the bottom-most case is sufficiently similar to its unlabelled 3 nearest neighbours but also farthest away from its labelled 3 nearest neighbours, and so would be a good candidate case for selection.

5.1 Cluster Utility Score

We first consider the problem of choosing the most interesting cluster. We look at how distances between cases and information gleaned from both labelled and unlabelled cases can be exploited when formulating a Cluster Utility Score *CIUS*.

For a given case c in a cluster C , our first metric estimates the average distance $distance_N$ between c and its neighbourhood of k nearest neighbours in C .

$$distance_N(c, C, k) = \frac{1}{k} \sum_{n \in N_k(c, C)} distance(c, n)$$

where $N_k(c, C)$ returns the k nearest neighbours of c in C , and $distance(i, j)$ is the normalised distance between cases i and j .

The simplest utility score for cluster C is the average $distance_N$ for the cases in the cluster. However, we further wish to influence this by the error associated with the incorrectly classified labelled cases in the leaf node of the decision tree corresponding to the cluster C . The intuition behind this is that if two clusters have the same score then the cluster containing cases with different class labels should be chosen over a cluster containing cases with similar class labels. Therefore *CIUS* combines the average neighbourhood distance $distance_N$ and the entropy of C 's subset of labelled cases L_C .

$$CIUS(C) = \frac{entropy(L_C)}{|C|} \sum_{c \in C} distance_N(c, C, k)$$

where *entropy* is the standard information theoretic measure

$$entropy(L_C) = - \sum_{i=1}^m \left(\frac{l_i}{|L_C|} \right) \log_2 \left(\frac{l_i}{|L_C|} \right)$$

m is the number of classes, and l_i is the number of cases in L_C belonging to class i .

However in a decision tree with nominal attributes, a leaf node may contain no labelled cases (e.g. cluster (ii) in Figure 4). Entropy is meaningless because L_C is empty,

but yet this is an interesting cluster for case selection. To overcome this problem we increment the class counts for each cluster by one. Thus l_j becomes $l_j + 1$, and $|L_C|$ becomes $|L_C| + m$. The revised definition for *entropy* follows and is used in the definition of *CIUS* above.

$$entropy'(L_C) = - \sum_{i=1}^m \left(\frac{l_i + 1}{|L_C| + m} \right) \log_2 \left(\frac{l_i + 1}{|L_C| + m} \right)$$

Let us demonstrate the effect in a binary classification domain. If a leaf contains only unlabelled cases then the revised entropy for this cluster is 1 ($\log_2 2$). However if there are labelled cases for each class, say 6 positives and 1 negative, then the class counts will be updated to 7 positives and 2 negatives, and the cluster entropy will be reduced from 1 to 0.76.

5.2 Case Utility Score

While *CIUS* captures the uncertainty within a cluster, the case-utility-score *CaUS* captures a case's impact on refining the case base index. The decision nodes that are traversed in order to reach a leaf node are chosen because of their ability to identify labelled cases with similar retrieval behaviour, by discriminating them from the rest of the labelled cases. Essentially the cases in a cluster share a common traversal path, and those that are likely to cause a change are unlabelled cases that are least similar to labelled cases in the cluster. However we would also like to ensure that selected cases are representative of any remaining unlabelled cases in the cluster.

CaUS is calculated for a case c in a selected cluster C by calculating the distances to remaining labelled cases in L_C and the unlabelled cases in the cluster U_C . Since we are only interested in selecting unlabelled cases labelled cases will have a *CaUS* score of zero. The diversity measure (adapted from [14]) assigns higher scores to cases that are farthest away from labelled cases, but also favours cases that are part of a tightly knit neighbourhood of unlabelled cases. Essentially it attempts to address the trade-off between selecting labelled cases that are not too similar to already labelled cases in the cluster, yet ensuring that they are sufficiently similar to unlabelled cases, thereby representing a higher proportion of unlabelled cases in C .

$$CaUS(c, C) = \begin{cases} 0 & \text{if } c \in L_C \\ diversity(c, C) & \text{otherwise} \end{cases}$$

$$diversity(c, C) = distance_N(c, L_C, k) * (1 - distance_N(c, U_C, k))$$

5.3 Sampling Heuristics

Let us revisit Figure 1 and see how the different steps fit together. **create-domain-model** uses available labelled cases to derive a model; here the case base index. This model is used to partition all the labelled and unlabelled cases to form **clusters**. *CIUS* is calculated for each cluster. These scores identify **K-clusters** from which

we select cases based on their *CaUS*s. Case selection is incremental, in that once a selected case is labelled, the L_C and U_C are updated, before another unlabelled case is selected.

Several incremental sampling techniques have been implemented with the more informed sampling techniques employing *CIUS* and/or *CaUS* for cluster and case selection.

- RND selects a cluster randomly and selects cases randomly (without replacement)
- ;
- RND-CLUSTER selects one cluster randomly and incrementally selects the cases with highest *CaUS* from this cluster;
- RND-CASE selects the one cluster with highest *CIUS* but selects cases randomly;
- INFORMED-S selects a single cluster with highest *CIUS* and incrementally selects highest *CaUS* cases from this cluster;
- INFORMED-M selects K (multiple) clusters ($K=3$) with highest *CIUS* and selects the case with highest *CaUS* from each cluster. Notice here case selection need not be incremental, hence the L_C s and U_C s are updated once in a single sampling iteration.

With all techniques in each iteration of the sampling loop unlabelled cases are incrementally selected until a sample of `Max-Batch-Size` is formed. We use RND as the technique with which to compare the more informed selection techniques. RND-CASE and RND-CLUSTER demonstrate the impact of *CIUS* and *CaUS* independently. INFORMED-S and INFORMED-M both use *CIUS* and *CaUS*, and should be better able to pick useful cases compared to RND-CLUSTER and RND-CASE. However it is harder to postulate the impact of selecting from a single cluster versus multiple clusters; it is likely that this is domain dependent.

6 Evaluation

We evaluated the different case selection techniques to determine whether informed case selection leads to case bases with increased accuracy. We selected six datasets from the UCI ML repository [2]. They have varied number of features, number of classes, proportion of nominal to numeric features, and some have missing values. In order to simulate similar problem-solving experiences in the case base the size of each dataset was doubled by randomly duplicating cases. The intuition behind this is that an informed sampling technique would avoid selecting similar problem-solving experiences that are already covered by the labelled cases in the case base. Two of the domains have in excess of 400 cases (House votes and Breast cancer). For these, a training set of 350 is randomly selected while the remaining cases form the test set. For the smaller domains (Zoo, Iris, Lymphography and Hepatitis), we formed a training set size of 150 and a disjoint, similarly sized, test set. Experiments with five increasing training set sizes were carried out starting at:

- 150 with an increment of 50 for the larger training sets; and
- 50 with an increment of 25 for the smaller training sets.

Although all cases in the training set are labelled for experimentation purposes, these labels are ignored until cases are selected from the training set. The labelled cases (L) forming the case base is initialised by selecting 35 cases from the training set (we use 15 for the smaller domains), the rest of the cases form the set of unlabelled cases (U). The sampling process terminates once 3 sampling iterations are completed, simulating a sampling process constrained by expert availability. The `Max-Batch-Size` is set at 3 which generally gives a selection technique the opportunity to select not more than 9 new cases for the case base.

The experiments aim to evaluate the effectiveness of informed case selection on case base accuracy and the efficiency on sampling time. We note the time taken to complete the sampling iterations. The accuracy of the case base, now with the newly sampled cases, is evaluated on the test set by a straightforward k -NN. The graphs plot the average accuracy over 25 trials for increasing training set size.

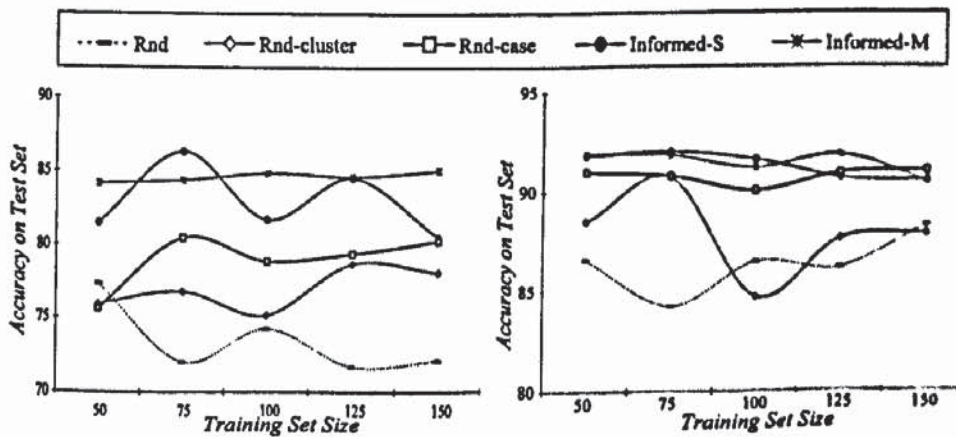


Fig. 5. Results for: (a) Zoo (b) Iris

6.1 Accuracy Results

Figures 5 (a) and (b) show average accuracy results for the Zoo and Iris domains. Both have no missing values, while Zoo has 7 classes and 18 features (all nominal), Iris has 3 classes and 4 features (nominal + numeric). As expected we see a significant difference between informed selection techniques INFORMED-S and INFORMED-M compared to RND (Zoo $p=0.001$ and Iris $p=0.001$). Although with Zoo we see that the difference between informed techniques and RND increase with increasing training set size this is not obvious with Iris. This might be explained by the contrasting difference in average index complexity (or number of nodes): 3 for iris and 8 for Zoo. When comparing INFORMED-M and INFORMED-S we see that the increase achieved by the former is significantly higher for Zoo ($p=0.016$), but that there is no difference for Iris ($p=0.371$). Again this is related to the relative difference in concept complexity, hence the difference in the number of partitions or clusters. It also suggests the need to consider

more than one cluster and possibly the inter relationship of clusters when selecting cases for domains with flatter and broader indices such as Zoo. The performance of RND-CLUSTER is less consistent compared to RND-CASE. This is interesting because it confirms that working hard at selecting a cluster is important.

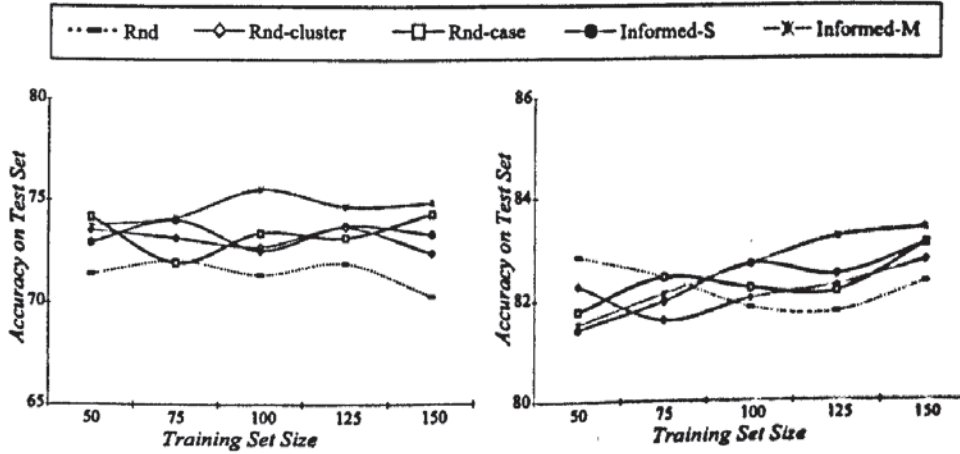


Fig. 6. Results for: (a) Lymphography (b) Hepatitis

Evaluation results for Lymphography and Hepatitis domains appear in Figures 6 (a) and (b). These domains have similar number of features (19 and 20), but Lymphography has nominal and numeric with 4 classes, while Hepatitis is binary classification with nominal features. Additionally the Hepatitis dataset contained missing values and when building the index a fraction of a case with missing values passes down each branch of an unknown decision node. For clustering, this means that a case with missing values can end up in two or more clusters. Here, we assigned cases with missing values to a single leaf; i.e. the leaf associated with the highest case fraction.

Again we find a significant difference between the informed techniques and RND (Lymphography $p=0.001$ and Hepatitis $p=0.001$). With Lymphography the average concept complexity is 9 and, as with Zoo, INFORMED-M's performance is significantly better than INFORMED-S ($p=0.005$). With certain test runs RND-CASE has outperformed INFORMED-S suggesting that random case selection is better than selection based on *CaUS*. We believe that there can be a danger of exploiting information from already labelled cases where selection can be too biased by labelled cases in the cluster; here with *CaUS* the distances to labelled cases $distance_N$, may become an undesirable influence when the distribution of labelled cases is skewed, possibly explaining the improved performance of random case selection. With Hepatitis the increase in accuracy by the informed techniques over RND though significant is small (2%) compared with other domains. It seems that the initial selection of labelled cases forming the case base already achieved high accuracy, therefore the impact of the newly sampled 9 cases seems to be less obvious.

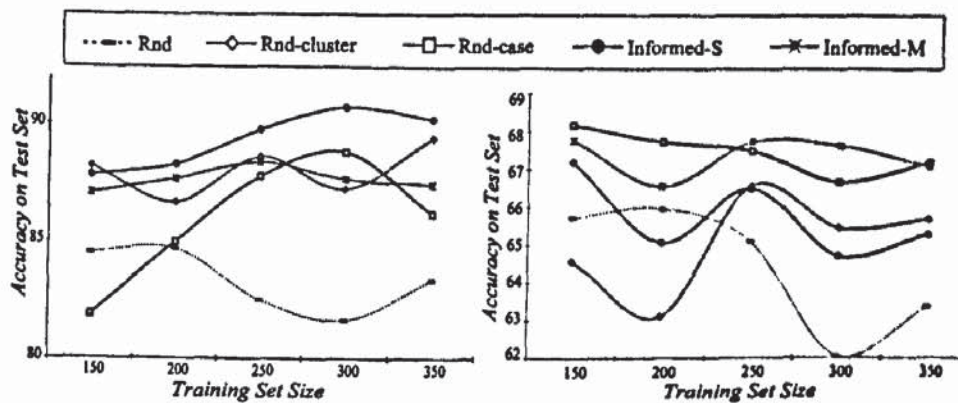


Fig. 7. Results for: (a) House votes (b) Breast cancer

The results from the larger datasets; House votes and Breast cancer appear in Figures 7 (a) and (b). These are both binary classed domains containing missing data. House votes has 16 binary valued features while Breast cancer has 9 multi-valued features. With House votes we have a significant improvement with INFORMED-M and INFORMED-S over RND ($p=0.001$). The concept complexity with House votes on average is 3 which suggests fewer partitions of the case base explaining INFORMED-S's significant improvement over INFORMED-M's performance ($p=0.023$).

Unlike the House votes domain, Breast cancer is more complex because the features are not just binary valued. Still the results are very encouraging, in that we have significant improvements with all informed techniques. We also see that RND-CASE has significantly better performance over INFORMED-S and RND-CLUSTER. This is most likely due to the increased concept complexity (here an average of 7, with a depth bound of 5 on the index tree) and so increased number of partitions to the case base. With fewer labelled cases in a cluster, *CaUS* would have been overly influenced by the labelled cases particularly in $distance_N$. This then suggests that for *CaUS* we need to consider the distribution of labelled cases and use this information to regulate the influence of labelled cases on *CaUS*.

6.2 Efficiency Results

So does the increase in accuracy justify the increase in training time? For domains with many features, and in particular those with many classes (e.g. Zoo), there is almost a 3-fold increase in training time with INFORMED-M compared to RND (see Table 1). This increase can amount to as much as a 40-60 seconds difference and for the larger datasets with 350 cases a 5-fold increase can mean up to a 150 second difference. For real applications this is an obvious drawback. However since the main processing cost is associated with the pairwise case distance calculation associated with *CaUS*, an efficient feature subset selection technique will help improve efficiency.

Generally when operating with a fixed expert availability time constraint the trade-off here depends on whether during this time we wish to present an expert with: fewer

Table 1. Increased training time ratio of INFORMED-M compared to RND

Training set size	50	75	100	125	150
Zoo	1.1	1.6	2.3	2.6	2.9
Iris	1.5	1.7	1.9	2.1	2.3
Lymphography	1.5	1.9	2.1	2.4	2.6
Hepatitis	1.9	1.9	2.1	2.1	2.3

Training set size	150	200	250	300	350
House Votes	2.3	2.8	3.4	3.9	4.5
Breast Cancer	2.6	3.3	4.1	4.4	4.7

yet different problems selected using informed techniques; or many problems selected randomly with the hope that a sufficient spread of problems is covered.

7 Conclusions

The idea of exploiting the partitions formed by the case base index as the basis for selective sampling for CBR is a novel contribution of this paper. It is also a sensible thing to do because the index invariably captures the CBR system's problem solving behaviour. The sampling approach is iterative and attempts to identify new cases that when added into the case base are most likely to trigger refinements to the index. The paper introduces a cluster utility score *CIUS*, which reflects the uncertainty within a cluster by deriving information from both labelled and unlabelled cases. High scores denote interesting clusters that are a source of useful unlabelled cases. A further case utility score *CaUS* then helps rank these cases by maximising distances to labelled cases yet minimising distances to unlabelled cases.

The effectiveness of informed sampling techniques using *CIUS* and *CaUS* was demonstrated on 6 public domain datasets. In general, a significant improvement in test accuracy was observed with these techniques compared to random sampling. Case selection from multiple clusters outperformed selection from a single cluster on several domains. However there seems to be an obvious relationship between index depth and breadth, and hence the partitions, and the sampling techniques. Generally for an index with fewer leaf nodes, selection from a single cluster works better but the opposite is true for a flatter index.

In this paper we have primarily concentrated on a case base index created by a decision tree structure formed using C4.5's information gain ratio. However, we are keen to see how the sampling approach presented in this paper might be more generally applied with other case base indexing schemes, such as k-means and bottom-up clustering. Another area of interest is to explore how incomplete case base indices might be manipulated as evidence of holes in the case base and then to exploit this as a means to discover new cases.

Selective sampling tools are useful for CBR systems whether labelled cases are plentiful or not. If there is a constraint on availability of labels then certainly a CBR

system with sampling capability will be very attractive. Conversely if there are many labelled cases then sampling techniques can be adapted as a means to identify few yet useful cases thereby ensuring that CBR retrieval efficiency is maintained.

References

1. Aha, D.W., Breslow, L.A.: Comparing simplification procedures for decision trees on an economics classification task. Technical Report AIC-98-009, Navy Center for Applied Research in AI, Washington DC, 1998
2. Blake, C., Keogh, E., Merz, C.: UCI repository of machine learning databases, 1998. <http://www.ics.uci.edu/mllearn/MLRepository.html>
3. Cohn, D., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996
4. Craw, S., Wiratunga, N., Rowe, R.: Case-based design for tablet formulation. In *Proceedings of the 4th European Workshop on Case-Based Reasoning*, pages 358–369, Dublin, Eire, 1998. Springer
5. Cunningham, P., Bonzano, A.: Knowledge engineering issues in developing a case-based reasoning application. *Knowledge-Based Systems*, 12:371–379, 1999
6. Jarmulak, J., Craw, S., Rowe, R.: Genetic algorithms to optimise CBR retrieval. In *Proceedings of the 5th European Workshop on Case-Based Reasoning*, pages 136–147, Trento, Italy, 2000. Springer
7. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning: Proceedings of the 11th International Conference*, pages 148–156, New Brunswick, NJ, 1994. Morgan Kaufman
8. Lindenbaum, M., Markovich, S., Rusakov, D.: Selective sampling for nearest neighbour classifiers. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI 99)*, pages 366–371, Orlando, FL, 1999. AAAI Press
9. Liu, B., Wang, K., Mun, L.F., Qi, X.Z.: Using decision tree induction for discovering holes in data. In *Proceedings of the 5th Pacific Rim International Conference on Artificial Intelligence (PRICAI-98)*, pages 182–193, 1998
10. McCallum, A., Nigam, K.: Employing em in pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 359–367, 1998
11. McKenna, E., Smyth, B.: Competence-guided case discovery. In *Proceedings of the 21st BCS SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence (ES 01)*, pages 97–108, Cambridge, UK, 2001. Springer
12. McSherry, D.: Automating case selection in the construction of a case library. In *Proceedings of the 19th BCS SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence (ES 99)*, pages 163–177, Cambridge, UK, 1999. Springer
13. Smyth, B., Keane, M.T.: Remembering to forget. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 377–382, Montreal, Quebec, 1995. Morgan-Kaufmann
14. Smyth, B., McClave, P.: Similarity vs. diversity. In *Proceedings of the 4th International Conference on CBR*, pages 347–361, Vancouver, BC, Canada, 2001. Springer
15. Smyth, B., McKenna, E.: Competence models and the maintenance problem. *Computational Intelligence*, 17(2):235–249, 2001
16. Wiratunga, N., Craw, S., Rowe, R.: Learning to adapt for case-based design. In *Proceedings of the 6th European Conference on Case-Based Reasoning*, pages 421–435, Aberdeen, Scotland, 2002. Springer