Machine Learning for Automatic Classification of Remotely Sensed Data

Linda Milne

A dissertation submitted to the
School of Computer Science and Engineering
University of New South Wales, Australia
in fulfilment of the requirements for the degree of
Doctor of Philosophy

September 2008

ORIGINALITY STATEMENT

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Signed .....

Date .....

COPYRIGHT STATEMENT

Signed .....

Date .....

AUTHENTICITY STATEMENT

Signed .....

Date .....

For Mum

Abstract

As more and more remotely sensed data becomes available it is becoming increasingly harder to analyse it with the more traditional labour intensive, manual methods. The commonly used techniques, that involve expert evaluation, are widely acknowledged as providing inconsistent results, at best. We need more general techniques that can adapt to a given situation and that incorporate the strengths of the traditional methods, human operators and new technologies.

The difficulty in interpreting remotely sensed data is that often only a small amount of data is available for classification. It can be noisy, incomplete or contain irrelevant information.

Given that the training data may be limited we demonstrate a variety of techniques for highlighting information in the available data and how to select the most relevant information for a given classification task. We show that more consistent results between the training data and an entire image can be obtained, and how misclassification errors can be reduced. Specifically, a new technique for attribute selection in neural networks is demonstrated.

Machine learning techniques, in particular, provide us with a means of automating classification using training data from a variety of data sources, including remotely sensed data and expert knowledge.

A classification framework is presented in this thesis that can be used with any classifier and any available data. While this was developed in the context of vegetation mapping from remotely sensed data using machine learning classifiers, it is a general technique that can be applied to any domain. The emphasis of the applicability for this framework being domains that have inadequate training data available.

Acknowledgements

Firstly, I must thank Dr Andrew Skidmore, Dr Brian Turner and Dr Tom Gedeon for giving me the chance to work on such a wonderful project. In spite of my moaning it really has been worth it and I have learnt more than I can ever say. Tom deserves a special mention here as he stuck it out the whole way – three cheers and thanks for everything.

Thank you to Dr Geoff Whale, Professor Paul Compton and Ken Robinson for the thousands of little things – and not so little things – that you did to support me over the years. I can never repay you personally, but I hope I can help someone else the way you have all helped me.

John "Albanana", Dave Brunato and the team at the CSE help desk have been great friends and I love youse all. For Magda, one elephant, as promised.

Without the Spatial Analysis Unit at Charles Sturt University, Wagga Wagga this work would not have been possible. Their constant energy and enthusiasm kept me going at a critical time and will never be forgotten – of course the help with data has also been priceless. In particular, thanks to Dr David Lamb and Gary McKenzie.

Of course my family have got me where I am today – thank you Patricia Milne, Malcolm Milne, Barbara Milne and Chuckles. But really, this work is dedicated to the special up and coming generation of the Milne clan, Crystal and Cooper. I hope its been worth the wait.

Thanks to Bill Cruickshank and Joe Liske for not laughing at my stupid questions. And to Associate Professor Jim Franklin from the School of Mathematics who was there to help fill in the gaps as well as support and encouragement.

Charles Willock – what can I say??? Thanks for the constant nagging and *helpful* suggestions – you were often right and it taught me a lot.

To my study buddy, Martin De Groot. I can never thank you for your support

and our motivational coffee drinking sessions. You helped keep me going and it means more to me than I can ever say.

And last but not least, to my supervisor Dr Andrew Taylor – though it is not enough all I can say is thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis we investigate the analysis of remotely sensed data specifically in the context of mapping vegetation, particularly where limited information is available. Maps are developed to show where particular plants or plant communities are likely to occur for a given area.

The advantages of detailed and accurate vegetation maps are numerous, however, their generation remains an expensive and largely manual task. Technological changes have meant improvements in our ability to automatically generate such maps, some of which are developed and demonstrated in this thesis.

1.1 The Uses of Vegetation Maps

Vegetation maps can be used in a number of ways. Native forests still cover much of the Australian continent, as well as countries such as Alaska and Brazil, with significant areas essentially untouched by man. Any hope of protecting these areas lies with understanding and monitoring them.

The Australian National Forest Inventory aims to characterise Australian forests for the entire continent to enable better decision making [120]. Aside from the sheer size of such an area, large parts are inaccessible and so impossible to map using surveys. Similar problems exist in Alaska, with the need to monitor 151 million hectares of land [140] and, of course, the South American rainforest's [100]. Even if large scale surveys were possible it is an ongoing concern as natural envi-

ronments are constantly changing.

In 1995 the NSW state government initiated the Basincare project, an environmental planning initiative to control and restrict clearing of native vegetation as a means of conserving the existing biodiversity and preventing further land and water degradation [55]. As part of the Basincare project a state-wide vegetation map is being generated which will provide a basis for long term monitoring of vegetation.

Not only do we need to monitor natural environments we need also to monitor agricultural and plantation crops for irrigation, nutrient levels, plant stress and pest and weed infestations [153]. In this case, changes can be on the scale of weeks or months, rather than years as may be the case with forests. Damage can be rapid and costly, not only to individual farmers, but to a country as a whole through lost export dollars or an increased dependance on imports.

A low cost, fast and effective means of monitoring agricultural crops can ensure the reduction and better targeting of the use of pesticides, herbicides and synthetic fertilisers. This can result in reduced production costs for farmers, as well as reducing the environmental impact of modern agricultural techniques.

Agricultural crops are not limited only to those that are annually planted and sown. Sugar maple is of major economic significance in north-eastern America [180]. In Australia, perennial species that are of importance include citrus and grapes. Such crops could also be monitored for pests or appropriate nutrient and water levels.

There is a clear need for vegetation maps that will help us to monitor and manage our natural resources. The benefits of more accurate mapping techniques are both economic, environmental and social. To protect vegetation we need to know, not only where it is, but how much of it there is, what species exist and in what ways they are threatened. Over time we need to be able to distinguish between threats and natural variability of the system.

1.2 Difficulties in Vegetation Mapping

Generation of vegetation maps requires the characteristics of a given plant community to be established and then to monitor the changes that are occurring.

However, vegetation mapping using current techniques is costly and error prone.

Ground survey data has long been used to produce vegetation maps, but have a number of drawbacks. A small number of sites, relative to the size of the overall area being mapped, are likely to be used due to the costs and difficulties in surveying. This means that it is likely that isolated pockets of unique plant communities, as well as the changes due to natural variation and micro climates, will be missed. There are also the problems associated with human evaluation, such as inconsistent judgement and variation in expertise. Overall, this means that ground survey data is scarce and when available can be of an unknown or questionable accuracy.

Additional complexity in mapping tasks is introduced by the constantly changing nature of plant communities, requiring continual updates of maps. Survey data is usually collected over a long period of time and may even be held by a number of different organisations, with different data standards. Overall, map generation is a long term, highly subjective and costly process.

While survey data is expensive to collect, and subjective in nature it is still considered necessary to generate useful maps [26]. It can provide us with attributes that can be used to determine which species are occurring in an area being studied, such as known existing species, soil type, aspect and topography. Given that we have such data we should still use it for mapping applications, if we have ways of dealing with the subjectivity and inaccuracy.

The widespread availability of aircraft has meant it has been possible to collect photographs of the earths surface on a regular basis. Such data provides reasonable levels of detail for large areas, with photo-interpretation traditionally being used to generate vegetation maps. However, it has been found that these maps can also contain significant errors due to human evaluation [26].

Statistical techniques have also been used to generate vegetation maps. These techniques are most effective with numerical data, and incorporating non-numerical data, such as a subjective human evaluation, is difficult. However, such techniques can be used to produce reasonably accurate maps for specific data sets. The particular shortcoming of such methods is that they are not necessarily predictive over the large areas for which maps may need to be generated, and they require large amounts of up-to-date and accurate data to be most effective.

To further compound the problems with vegetation mapping, complete informa-

tion is difficult to collect. That is, all possible contributing factors that cause a particular plant species to grow in a specific area are not completely understood and so it is impossible to collect all the required data for accurate mapping. To simplify a mapping task, a reasonable set of attributes are identified that can be used to classify plant communities. Simplifications such as these result in the loss of information and potentially reduce the long term efficacy of the collected data.

As we are a long way from understanding plant communities and surveying techniques are inherently inaccurate, any maps we do generate will also contain, at least, some generalisations, and at worst significant errors. As more information comes to light it would be useful to have reliable and consistent ways of updating maps. This means that we need to have an idea of how well a given map represents an area and so have an understanding of when the changes observed are due to true changes or inadequacies in the original classification[1].

Given the above factors, the amount and quality of training data available for use in classification schemes is going to be small in comparison to the ground area to be classified. As human development is moving so quickly it is no longer possible or useful to generate and refine vegetation maps over long periods of time. Changes can be quick, and permanent damage can be done. Thus, we need ways to generate up-to-date vegetation maps quickly and efficiently, with as high a level of accuracy as possible.

1.3 Vegetation Mapping from Remotely Sensed Data

Remotely sensed data is any kind of data collected at a distance from the object of interest, including images of the Earth's surface from satellites and aircraft. These images can provide data for large areas on a regular basis, and allow monitoring without physical interference. Such data is used as the basis for generating vegetation maps.

One of the strengths of remotely sensed data is the large areas for which data is available, far more than could ever be effectively surveyed. The Landsat system,

[1]Map in the context of this work meaning a classification of an entire remotely sensed image. The terms map and classification are used interchangeably throughout this work.

for example, gives complete coverage of the Earth's surface every 18 days [134]. Consistent, long term data collection using remote sensing means that relationships can be found that might otherwise be missed if relying only on human evaluation skills and surveys. It also has the power to provide information that may be overlooked or under-estimated from ground level information [86].

Remotely sensed data can be utilised in the initial phases of a mapping task to support survey work. An initial investigation of a remotely sensed image for the area in question can help identify the most productive sites to survey as well as highlight areas of interest that may otherwise have been over looked.

It is widely accepted, however, that the use of remotely sensed data alone is not enough in vegetation mapping applications. The characteristics of vegetation, as represented in remotely sensed data, can change from day to day, and even hour to hour, based on factors such as moisture and nutrient levels, and atmospheric conditions. Individual plant species are identified not just by their spectral characteristics but also by bark colour and texture, the shape of the leaves and other information not currently available in remotely sensed data. Information about topographical location and climate are also important in determining which plant species will occur. So, while it is agreed that human interpretation and other forms of surveyed data are is fraught with problems, they can still be used to add value to the use of remotely sensed data.

With the increasing amounts of remotely sensed data the task of manually analysing it has become intractable. A database can contain gigabytes of data for one small study area. As higher resolution data becomes more readily available the problem will only compound.

In summary, to enable useful vegetation mapping from remotely sensed data we need more general techniques that can adapt to a given classification task and be able to process large amounts of data, as well as incorporating the strengths of the traditional classification methods and human operators.

1.3.1 Knowledge Based and Machine Learning Systems

Geographic information systems are used to store the types of data that are used to generate vegetation maps, including remotely sensed data. However, the ability to fully integrate remotely sensed data with such systems has been hampered by the

need for human interpretation and assistance [65]. Effective utilisation of human expertise, through knowledge based systems, can mean that we can extract the maximum amount of information from remotely sensed data and produce high quality vegetation maps.

Interpretation of individual spectral values given in remotely sensed images is difficult and not usually done by humans. Human interpretation usually involves subjective concepts such as texture, colour and shape. While the use of expert knowledge is useful in vegetation mapping, it is not necessarily clear when and how it can be most effectively used when classifying remotely sensed images.

A variety of systems have been developed that incorporate expert knowledge and the classification of remotely sensed imagery. While these systems have been successful for specific applications they still suffer from a problem called the knowledge acquisition bottle neck. That is, the difficulty in transferring the knowledge of an expert into a set of computer usable rules. This is particularly pronounced in remote sensing domains as there may not be an expert for a particular area due to its inaccessibility or the fact that it has not yet been studied.

Reasons traditional knowledge acquisition approaches are limited in remote sensing applications include:

- Experienced photo-interpreters can spend large amounts of time generating rules.
- The rules need to be updated for different geographical regions.
- No spatial rules exist for complex imagery.
- Limited amounts of ground survey data.
- Maintaining the consistency of the knowledge base.
- Increasing volumes of remotely sensed data.

While progress has been made in addressing these issues there still do not exist automated systems that show robust and accurate behaviour across a wide range of image data and ancillary[2] information.

[2]Ancillary information or data is a commonly used term in remote sensing to describe any data that can be used to enhance the information in spectral data for generating a map. This includes, but is not limited to, expert knowledge, climate data, topographic and soil data.

Recently the field of machine learning has developed in an attempt to address such issues. Machine learning refers to the set of algorithms which acquire knowledge through experience [170], and can be applied to a wide range of classification domains. Experience is generally provided in the form of a data set containing a number of cases with a known set of attributes used to describe them and with each case belonging to a known class. Machine learning provides us with a number of classification techniques that can be used to automatically generate classifications of a remotely sensed image and are able to incorporate the information from a wide range of other data sources.

The strengths of machine learning techniques include:

- The ability to use numeric and non-numeric data in a classification.
- The ability to use expert knowledge to direct or enhance classification.
- Some classifier systems can yield an explanation for a particular conclusion.

Due to the difficulties in collecting data in vegetation mapping domains all information that is available should be used. To do this we need to be able to incorporate data that is available from satellite imagery, from a number of platforms over time, expert knowledge and other mapped information such as soil and existing vegetation maps. Machine learning techniques provide us with a number of ways of doing this.

1.4 Contribution Demonstrated in this Thesis

The focus of this work is on domains that are difficult to apply standard classification techniques to, due to inadequate data being available. This thesis demonstrates techniques that allow the extraction of as much information as possible from all sources of available data, and then choosing only the most relevant information for the automatic generation of accurate classifications. Such classifications are generated by combining the results of a number of simple classifiers. The final outcome of this thesis is:

> *To provide a framework for automatically generating accurate classifications, using any classifier and any available data.*

While the framework described is general enough to be applied to any domain, the particular domain that is used to demonstrate its use is vegetation mapping from remotely sensed and ancillary data. Nel et al. [117] propose the notion that mapping need not be an exercise in providing accurate maps, rather it can be used to identify where particular features might be found. The classification framework described here also intends to meets this aim. We show ways to automatically generate maps that will provide both information about areas in which vegetation classes are most likely to occur and also to show areas that need further investigation.

Specifically, the new techniques developed that are incorporated into the classification framework are:

Highlighting information in the available data. As we are interested in domains for which small, noisy datasets are available for training classifiers we need to extract as much relevant information as possible from the data that is available. Techniques to achieve this are demonstrated.

Neural network attribute selection and performance improvement. Neural networks have been successfully demonstrated for a wide range of image analysis problems. While neural networks are able to distinguish objects in the presence of noise or irrelevant information, they can be made to perform better when such information is removed from the training data. A new method of attribute selection for neural networks was developed and successfully used. Heuristic methods for automating and improving the performance of neural networks are also demonstrated. Thresholding the output values of neural networks, to improve classification accuracy, was also developed and demonstrated.

Simulating a remotely sensed image. A technique for generating plausible remotely sensed images was developed. The simulated image is generated from the known properties of a real image. Most importantly, the image is generated in such a way that the classification of every pixel in the image is known. This allows us to compare classification techniques quantitatively and assess the correspondance between the error on the training data and the error over the entire image.

Combining a number of simple classifiers to improve classification accuracy. A technique for combining simple classifiers was developed. A classification is broken down into a number of simple tasks rather than training one large classifier to recognise everything. In this way we can improve classification accuracy,

and consistency. Firstly, a hierarchy of classes is generated. A number of simple binary classifiers are trained on different views of the data and a classification is given to a pixel only when all classifiers agree on class membership. The performance of individual classifiers is further improved by using attribute selection. This approach also means that we can produce more consistent results between the training data and the entire image, and so reducing the number of misclassified pixels.

Automated generation and assessment of classifications of images. A classification system was developed that allowed automatic generation of a large number of maps which were ranked according to their measured quality.

The particular impediments to automatic generation of maps that are addressed in this work are:

- Large scale investigation of the available data to extract as much useful information as possible.
- The flexibility of neural network classifiers, in the topology and training regime, and determining when a given network has generalised the characteristics of a particular dataset.
- Producing consistent classification results between the training dataset and an entire image so that automatic evaluation of the quality of the classifications can be done. If we can not do this we need to rely on human evaluation of the classified images.

1.4.1 Thesis Organisation

It is important to understand the organisation of this thesis at the outset. As this work involves the combination of a number of disparate techniques the initial chapters introduce these concepts, largely in isolation, and later chapters demonstrate their combination. Some work may seem irrelevant or unrelated, but needs to be seen in its full context. To maintain the readability of this work each concept is individually introduced and discussed before the overall classification framework is introduced.

Chapter 2 discusses previous work in classifying remotely sensed data and machine learning as well as giving an overview of the classification algorithms and

analysis techniques that will be used throughout the thesis. This chapter is only intended as a general overview to the field and further discussion of specific topics is given in the relevant section, as required. This approach has been used to aid readability, again, due to the large number of disparate subjects that are incorporated into this work.

Chapter 3 outlines the properties of the datasets used in this work.

Chapters 4 through to 9 are introductory work, discussing each of the individual techniques that will later be combined to automatically generate vegetation maps. Each chapter discusses a single concept and demonstrates its use. In isolation these chapters may not seem relevant or useful, however, each of the ideas discussed are brought together in Chapters 10 and 11.

Chapter 4 discusses misconceptions about the use of neural networks and how to improve classifications using them.

Chapters 5, 6 and 7 then go on to investigate techniques for the generation of additional attributes that highlight information in remotely sensed images and methods for selecting the most pertinent attributes for a given classification task.

Chapter 7 introduces a novel attribute selection technique, specifically for use with neural networks, that was developed in the course of this work. As this is a new technique this chapter diverges temporarily from the main theme of this thesis to demonstrate its effectiveness as a generic attribute selection technique.

Chapter 8 demonstrates a number of techniques for improving the reliability of classifications. In particular, the thresholding of neural network output values is a new idea developed as part of this work.

Chapter 9 outlines a technique for simulating remotely sensed data so that complete class information is available for an entire image and so allowing the classification techniques developed throughout this thesis to be quantitatively evaluated.

The work in the Chapters 10 and 11 are the culmination of the work discussed in the earlier chapters and demonstrates the results from a fully automated classification system. The classification system takes a set of remotely sensed images, generates the specified additional attributes and returns a set of classifications and their rankings. These are the critical chapters in demonstrating the results of the automated classification of remotely sensed images.

Chapter 10 investigates automating classification and evaluation using the simulated data. This chapter demonstrates that it is possible to generate classification schemes such that the error rates on the training data can be translated to the error rate over the entire classified image in a meaningful way.

Chapter 11 demonstrates the automated classification and evaluation of real images.

Chapter 12 discusses the key points of this work and the direction of future work and conclusions.

Appendix A contains a glossary of terms and abbreviations used in both remote sensing and machine learning.

Appendix B gives links to the publications that were generated from this work.

Appendix C contains a survey of the literature published recently in multi-strategy classification and attribute selection.

1.4.2 Publications

The work discussed in these papers is included in and referenced in the relevant sections of the thesis.

L.K. Milne, T.D. Gedeon, and A.K. Skidmore. Classifying dry sclerophyll forest from augmented satellite data : Comparing neural network, decision tree and maximum likelihood. In *Proc. 6th Australian Conference on Neural Networks*, Sydney, pages 160–163, February 1995.

L.K. Milne. Feature selection using neural networks with contribution measures. In *AI'95 Poster Proceedings*, Canberra, November 1995.

L.K. Milne and C. Willock. Comparison of two methods for increasing training set size for neural networks. In *AI in the Environment Wkshp*, Canberra, pages 89–94, November 1995.

L.K. Milne. Attribute selection in neural networks used to classify remotely sensed data. In *Visual Information Processing Wkshp*, Sydney, pages 21–26, December 1997.

L.K. Milne. Improving Classification Accuracy of Machine Learning Techniques applied to Remotely Sensed Data. In *Proc AI'98*, Brisbane, pages 26–37, July 1998.

Chapter 2

Previous Work

Since the 1970's remotely sensed data from satellite platforms has become more readily available. It has been used in many mapping applications, including elevation model[1] development, orthoimage[2] production, automated update and generation of topographic maps[3], land usage and vegetation inventories and automated feature[4] extraction [157, 152, 11, 149, 163]. However, much work is still needed for accurate and consistent mapping to be achieved. In this chapter we look at some of the work that has been done to this end.

2.1 Mapping Applications

Fast and accurate generation of maps of natural resources is essential. If remotely sensed data is to be used on a more regular basis for real-world applications we need to have a better understanding on how to use it most effectively.

Assessment of the amount and extent of damage is essential to manage agricultural crops of various kinds [180, 151, 86, 87, 13]. It is also important to have up-to-

[1] An elevation model is a representation of the height and shape of the Earth's surface.

[2] An orthoimage is generated by rectifying distortions in an image caused by variations in the height of the terrain. It contains pixels that are all to the same scale.

[3] Topographic maps contain natural features such as hills and rivers, as well as cultural features such as roads, bridges and railways.

[4] Features in this context mean any object that can be identified in an image. These include, but are not limited to, buildings, trees, cities, forests, and even more intangible objects such as snow removal routes.

date inventories of forest resources for management and conservation purposes. Inventories have been, and continue to be, derived from human experts using aerial photographs and manually collected ground truth data [140]. Generating such maps is a difficult task, and is further complicated by the need to regularly update them to ensure appropriate decisions can be made [49].

The use of remotely sensed data in water management [118] and estimation of rainfall [109] has also been investigated. Again, to be of the most use maps need to be updated regularly.

Isolated and difficult to access forests are an obvious application of remotely sensed data. McGowen et al [103] look at monitoring the Jemalong area in Central Western NSW for water logging and salinity. The Australian National Forest Inventory aims to characterise Australian forests, across the entire continent, to enable more informed decision making [120]. Both these areas, and others around the world, are large enough that manual generation and update of maps is not feasible.

The major advantage of remotely sensed data in mapping exercises is that the data available covers large areas and can be collected on a regular basis.

2.2 Remotely Sensed Data

Many remote sensing platforms exist producing a number of different products, a sample of which can be seen in Table 2.1. Each platform has a particular type of sensor that takes reflectance measures for a set of predefined spectral ranges. Each spectral range is called a band. If the range of values is large the data is referred to as broad band, and for small ranges, narrow band. For example, recording values in a 2μm range of the electro-magnetic spectrum is a broad band measurement, while a range of 0.1μm is narrow band.

Each remotely sensed image that is acquired covers a particular area on the Earth's surface and contains a set number of spectral bands. Each band of a given image is essentially an image in its own right. The pixels in each of these images correspond to the reflectance measured from a particular point on the Earth's surface for the given spectral range. The spatial resolution for a given image is determined by the area on the ground that each pixel corresponds to. Data is low resolution

if each pixel in the image corresponds to a large area on the ground, while high resolution data corresponds to a small area.

Platform	Resolution	Wavelength(μm)
AVHRR	1.1km	1. 0.58-0.68 2. 0.725-1.1 3. 3.55-3.93 4. 10.3-11.3 5. 11.5-12.5
Landsat MSS	79m	4. 0.5-0.6 (green) 5. 0.6-0.7 (red) 6. 0.7-0.8 (near infra-red) 7. 0.8-1.1 (near infra-red) 8. 10.4-12.6 (thermal)
Landsat TM	30m 120m	1. 0.45-0.52 (blue) 2. 0.52-0.6 (green) 3. 0.63-0.69 (red) 4. 0.76-0.9 (near infra-red) 5. 1.55-1.75 (mid infra-red) 7. 2.08-2.35 (mir) 6. 10.4-12.5 (thermal)
SPOT (multi-spectral) (panchromatic)	20m 10m	0.5-0.59 0.61-0.68 0.79-0.89 0.57-0.73
ABVS	1m/2m	450nm (blue) 550nm (green) 650nm (red) 770nm (near infra-red)

Table 2.1: Selection of remote sensing platforms.

The most readily available remotely sensed data is AVHRR, due to its relatively low cost and the regularity with which the satellite returns to the same point on the Earth's surface. However, it is very low resolution data and so limited in its use. When using remotely sensed data there is a trade-off between the coverage[5], resolution and the cost.

More recently airborne video systems have been under development, using aircraft instead of satellites for collection of data. The advantages of such systems are that

[5]Coverage refers to how often a satellite passes over the same place on the Earth's surface.

images can be collected as often as required with very high resolutions, currently up to 1m, for a fraction of the cost of satellite data. In addition the advantage of airborne video systems over satellite systems is that it is easier to modify the spectral bands for which data is collected. Airborne video imaging is of particular use in agricultural applications due to the increased flexibility of data collection and higher resolutions. The ABVS system listed in Table 2.1 is a specific instance of an airborne video system, which will be discussed in more detail later.

In spite of the advances in data collection techniques, it is widely accepted that remotely sensed data alone is not enough for many applications. Incorporating other types of data, such as climate and topographic data, into classifications using spectral data has long been recognised as vital for improving classification performance [158, 59, 66, 103, 149].

2.3 Remotely Sensed Data for Mapping Applications

In spite of the advantages of remotely sensed data, there are a number of reasons why remotely sensed data can not be used for mapping applications in isolation. Distinct objects can look spectrally similar due to the mixing of spectra[6] of the objects, natural variability in spectral response and different objects having similar spectra.

The AVIRIS system was developed to enable better differentiation of different objects. It provides 224 narrow band spectral measurements per pixel, with a spectral resolution of 0.01μm [130]. However, work investigating the mapping of significantly different tree species has shown that distinctions are not always possible even with this level of spectral information [130]. That is, we require more than just remotely sensed data to generate accurate maps.

Before the widespread availability of fast, cheap computer based classification

[6]For lower resolution data, in particular, it is desirable to determine what is contributing to the spectral characteristics of each pixel in an attempt to provide reliable classifications. Pixels are usually treated as pure elements, that is a pixel contains a single clearly identified object. Use of the spectral characteristics derived in this way will lead to errors in classification. This is a significant area of study in its own right and will not be investigated here. Techniques for dealing with mixed pixels can be found in [58, 102].

systems the only solution for interpretation of images was using human expertise. When producing maps a human photo-interpreter utilises a large amount of prior knowledge that extends beyond the images being used and the context they are used in [40]. For example, when producing a vegetation map a photo-interpreter will use remotely sensed data, topographic, climate and soil information, as well as knowledge about the vegetation of the area and its physical characteristics. That is, a human expert will use all available information to identify patterns.

Ideally mapping should be based on remotely sensed data and be augmented with other types of data appropriate to the type of map to be produced. Specifically in the case of vegetation mapping, the types of data would include climate, topographical and soil data, as well as expert knowledge on the characteristics of the species being mapped.

Two specific areas of vegetation mapping that are of interest are agriculture and forestry. These will be discussed in more detail in the following sections.

2.3.1 Mapping for Agriculture

Remote sensing has been applied to problems such as detecting rust in wheat, blight in potatoes and corn, scale in citrus and root rot in field beans, and irrigation scheduling. Early identification of problems can result in containment, reduced need for chemical solutions and increases in yield. An indicative sample of remotely sensed data being applied to agricultural applications is given in this section.

Smith and O'Neill [150] investigated the identification of noxious weeds in pastures from spectral information. This initial work was to determine the spectral characteristics using an infrared spectro-radiometer, with the ultimate aim of being able to identify the weeds in remotely sensed images. The results indicated that sufficient information can be extracted from the airborne Daedelas scanner imagery[7] to be able to target areas of infestation for the specific weed species investigated.

Steven et al. [153] investigated the use of high spectral resolution data to monitor crop stress and so estimate productivity of a given area. It was hoped that the

[7]See [134] for more information on this platform.

increased spectral resolution would allow for better detection of changes in plant spectra. However, it was found that plant stress is a complex and little understood problem, and that the spectral characteristics of plants are only one of many indicators. Even if high resolution data of this type were inexpensive and readily available, useful monitoring is still limited by the long return times[8] and limited data for training classifiers.

Changes in conditions of agricultural areas can be fast and action may need to be taken within a day or so of problems occurring. Existing commercial satellite systems do not have the required resolution or return periods for reliable mapping of agricultural crops [116]. Currently the highest resolution is the SPOT sensor at 10m (in panchromatic mode only). Though higher resolution sensors are becoming available they will still provide less than ideal solutions due to inadequate coverage and high cost.

A purpose built airborne video system (ABVS) developed at the Spatial Analysis Research Unit at Charles Sturt University has been used in a number of agricultural applications [87]. The main advantage of this system, and others like it, are the high resolution of the data and the regularity with which data can be collected. Additional flexibility, over satellite imagery, is given as the spectral bands for which data is collected can easily be modified for specific applications by using different filters on the cameras used to collect the data. Results have so far been encouraging, and with the increased flexibility and reduced cost of data collection, is a solution that may be preferable to other types of remotely sensed data for this domain.

Pearson et al [124] investigated real-time monitoring of a variety of crops using airborne video. Farmers included in the study reported that images needed to be delivered in less than 48 hours to be of use. However, the frequency of coverage varies during the season from weekly to every two days. This study demonstrated that real-time monitoring was indeed possible for a number of agricultural applications, but that automated processing is the main concern in using the technology.

Lamb [86] demonstrated the use of airborne video for mapping weeds in cultivated fields. Maps such as these have the potential to allow targeted spraying for weeds, increasing the efficiency of herbicide usage from 2% to 60% of the herbicide reach-

[8]The return time of a satellite refers to the time it takes for it to return to the same location on the earths surface. The higher the resolution of the data, the larger the amount of data collected and the longer the return times.

ing the target. It was noted that ground inspection of fields provides an average picture of crop status, but intra-field variability is what makes management difficult and can result in substantially reduced yields. Airborne video data can go a long way to addressing these problems by providing detailed information on crop variation across a field. This in combination with GPS on farm machinery can be used to target specific areas within fields.

Another advantage of remotely sensed data over human evaluation is the ability to provide data outside the range of human perception. However, further work needs to be done to determine which spectral responses correspond to specific crop problems.

Monitoring of agricultural crops via remotely sensed imagery has been shown to be possible and even desirable. As data may be required as often as every two days, airborne video data is a better alternative to satellite imagery. Of course, weather and other factors can still prevent data collection with the desired frequency or quality. Mapping accuracy can be improved when remotely sensed data is augmented with other types of data, such as climate, soil and rainfall. However, airborne video monitoring of agricultural land is a relatively inexpensive and effective solution that has the potential to provide data as required if we can find the ways to automatically process it quickly and accurately.

2.3.2 Mapping Tree Species

Mapping tree species has slightly different requirements to agricultural mapping, but the implementation suffers from many of the same problems. Changes are generally over longer periods of time and the areas to be mapped are often significantly larger and less homogeneous.

The dominant native tree species in Australia are eucalypts whose characteristics are quite different to those of forests in the northern hemisphere, where much of this research is done. The areas for which maps are required can be vast making this a major undertaking, and remotely sensed data has a significant role to play. However, a lot more work needs to be done, under Australian conditions, to determine the best approach to effective use of this data [122]. With the amount of remotely sensed data available and limited amounts of ground truth data there is a need for techniques that will direct and support this type of investigation.

Ripple [135] used AVHRR data to determine the proportion of closed canopy in coniferous forests, as well as trying to characterise the spectral signatures of various successional stages. Although individual areas of clear-cut forest are not visible at 1km resolution there are changes in the reflectance for pixels containing such areas.

Similar problems were described by Caicco et al [18]. Landsat MSS data was used to map 71 vegetation and land-use categories for the state of Idaho, U.S.A. – an area of over 200,000km^2. Mapping accuracy was estimated to be around 92%. The conclusion reached was that the use of remotely sensed data gave an efficient means of assessing the protection of land-cover types and biodiversity over large areas. Limitations identified included the low resolution of the data being used and the lack of data on the ecological condition of vegetation complexes. That is, the remotely sensed data could be used more effectively with the incorporation of other types of data.

Stone et al. [154] developed a land cover map of South America, also using AVHRR data. Classifications using vegetation indices (see Section 5.2.3) were possible with an overall accuracy of around 90%. However, a problem with this work was the misclassification of sites in the study area and the variable reliability of the results across different classes.

Overall low resolution data can only be used to map broad characteristics, but this should not stop its use if it is the only available data. Techniques need to be developed that can make best use of what is available.

Wilkinson et al. [175] investigated the use of multi-source data to monitor forest ecosystems. The problem of mapping such environments is the complex mixtures of species. With the use of both Landsat TM data and SAR data mapping accuracies of 70-80% were able to be achieved. The use of the SAR data enabled better distinction between the broad leaf and conifer species. This particular study contained areas of eucalypts and increased accuracies in identifying these were found when both types of imagery were used. The problems with this type of work are the use of different resolution data, which requires re-sampling and registration. This pre-processing of the data can result in information loss and so decreased mapping accuracy.

Schreuder et al. [140] investigated the use of a four-phase sampling technique using Landsat MSS, colour infrared and ancillary data to inventory Alaskan forests.

While this technique had been used successfully in previous work [91], it did not generalise well to new study areas. Part of the reason for the failure of this technique was attributed to the commonly experienced problems of noise in the available data and not enough ground truth data.

Airborne video has also been used for forestry applications. It can provide detailed information, down to individual trees, for a fraction of the cost of satellite imagery or ground surveys. In addition, it can be collected as often as required.

Yuan et al. [180] successfully used airborne video data to assess the decline of sugar maple due to problems such as pollution and soil deficiencies. But again, the recommendation was made that additional types of data be used to improve the process.

Logging of native forests is of particular concern in Australia. Wood chips provide significant export income, and there is plenty of room for expansion in the market place. However, there is concern about the degradation of currently logged forests, let alone the concerns about expanding the industry further. Squire [151] discussed the need to balance sustained wood production and ecosystem conservation in native forests. In particular, Squire discusses the need for increased scientific research and monitoring of native forests. An important part of such evaluations are to include information such as political, financial and technological data. In this situation, and others like it, the use of pure statistical techniques are not necessarily appropriate as the data contain non-numeric attributes.

The obstacles to adopting widespread remote monitoring are common to all domains. Problems include insufficient data, coverage, registration, calibration, effective use of ancillary data and automated processing.

2.4 Classification Techniques

Much work has been done using statistical techniques for classifying remotely sensed data, an overview of which can be found in [145, 148]. However, as already noted, these techniques are not always adequate for dealing with small noisy datasets, nor can they easily be used with the ancillary data types that are required in mapping applications. Machine learning is an area of research that has more recently appeared on the scene and provides solutions to some of the problems

faced when using statistical techniques.

This section gives and overview of the specific techniques used in this thesis. We discuss their advantages and disadvantages, and the reasons for their use in this work.

2.4.1 Maximum Likelihood Classification

One of the most commonly used statistical classification techniques in remote sensing domains is maximum likelihood classification, which is based on Bayes theorem [134]. It is available in most remote sensing and image processing packages and is often used as the standard against which other classification algorithms are measured [74].

Classification is based on Bayes rule as follows. The n spectral values associated with each pixel can be written as a n-dimensional vector $\mathbf{x}_j$, for $j = 1..K$ pixels. For each class c_i and vectors $\mathbf{x}_j$ in that class, we calculate the mean vector and covariance matrix as follows.

$$\mathbf{m}_i = \frac{1}{K}\sum_{j=1}^{K}\mathbf{x}_j$$

$$\mathbf{C}_i = \frac{1}{K-1}\sum_{j=1}^{K}(\mathbf{x}_j - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_i)^t$$

where $\mathbf{v}^t$ is the transpose of vector $\mathbf{v}$.

The discriminant function for each class c_i is then given by

$$g_i(\mathbf{x}_j) = -ln \mid \mathbf{C}_j \mid -(\mathbf{x}_j - \mathbf{m}_i)^t\mathbf{C}_i^{-1}(\mathbf{x}_j - \mathbf{m}_i)$$

The class c_i assigned to a vector $\mathbf{x}_m$ is

$$\mathbf{x}_m \in c_i \text{ if } g_i(\mathbf{x}_m) > g_k(\mathbf{x}_m) \text{ for all } i \neq k$$

This assumes that the probability distributions of the data are multivariate normal. However, this assumption can cause significant errors, the extent of which is not known as the underlying probability distribution is usually not known [74]. One study found that nearest neighbour classification was more accurate and far more robust than maximum likelihood classification [74].

Training a maximum likelihood classifier requires a reasonably large sample of data – around $10n$ cases, where n is the number of attributes. Unfortunately, it is usually difficult to obtain this amount of data [134].

Unsupervised maximum likelihood classification can also be carried out by first generating a set of classes using an unsupervised classification technique, such as ISOCLASS [9]. Unsupervised maximum likelihood classification is used here as a data mining technique rather than a classification technique (see Section 2.4.10).

The algorithm maxlik, from the GRASS GIS package [169], was used for this work.

2.4.2 Problems with Statistical Techniques

Statistical techniques, including maximum likelihood classification, can be used with great success when only considering remotely sensed or other numerical types of data. It is also widely acknowledged that poor classifications will result from insufficient and missing data [134]. However, the biggest disadvantage to using statistical techniques is the difficulty in incorporating non-numeric data in a classification in a meaningful way. When we add this to the large amounts of training data required, we need to consider using other classification systems.

The Statlog project [108] investigated classification procedures on large-scale and commercially important problems. The aim was to determine to what extent these techniques met the needs of industry. Around 20 procedures were tested on 22 datasets, one of the datasets being satellite imagery.

The most effective five algorithms for classifying an image were nearest neighbour classification, learning vector quantisation (LVQ), DIPOL92[9], radial basis function

[9]DIPOL92 is a learning algorithm that constructs an optimised piecewise linear classifier [137]. Initially, discriminating hyper-planes are determined by pair wise linear regression. An error function is then defined based on the misclassified patterns. This function is then

network (RBF)[10] and ALLOC80[11].

The outcome of the Statlog Project was that traditional statistical techniques are not necessarily appropriate for classification of remotely sensed data. Rather, non-parametric techniques are more appropriate. Commonly used techniques that fall into this category, that were used in this work, are neural networks, decision trees and nearest neighbour classification.

2.4.3 Error Back-Propagation Trained Neural Networks

Neural networks are classifiers that aim to simulate the behaviour of the brain. They consist of simple nodes, that behave similarly to the neurons in a brain, that are inter-connected to mimic the complex behaviour of the brain.

They are widely regarded as a powerful image processing classification algorithm. The most commonly used and best understood networks are multi-layer perceptrons trained with error back-propagation [138, 96]. These types of networks were used in this work.

Each node in a multi-layer perceptron consists of n inputs and an output o, as shown in Figure 2.1. A node is a simple functional unit that computes the weighted sum of it inputs, with the output being some function of this sum. Typically the sigmoid function is the function applied to the sum to give the output of a node and to constrain the output values, defined as follows.

$$a_i = \frac{1}{1 + e^{-s_i}}$$

where s_i is the weighted sum of the inputs to the node and a_i is the output value, or the activation, of node i.

---

minimised using gradient descent.

[10] A radial basis function network [137] is similar to a multi-layer perceptron (MLP) but the hidden layer nodes compute an arbitrary function of the inputs (often Gaussian) and the transfer function of each output node is the identity function. Though RBF and MLP are computationally equivalent the RBF has some advantages over the MLP. The RBF does not suffer from finding local minima and is better able to make statements about the accuracy of the probabilistic interpretation of the outputs.

[11] ALLOC80 is a specific implementation of a non-parametric classifier that aims to estimate the underlying density function [114].

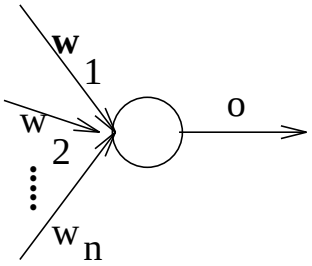


Figure 2.1: A node of a multi-layer neural network.

Any number of nodes can be connected in any number of layers to give a network, an example of which is shown in Figure 2.2. The first layer of nodes does not perform any calculation, simply distributing the input values to the next layer of nodes. Not every node in a layer need be connected to every node in the succeeding layer, though for this work all nodes are fully connected to all nodes in the next layer.

Each iteration, or epoch, in training the neural network presents each case in the training dataset to the input nodes and generates output values in the final layer. The total error for the dataset is given by:

$$E = \sum_{p} \sum_{i} (t_{pi} - o_{pi})^2$$

where p ranges over the set of input patterns and i over the number of outputs, t_{pi} is the target output and o_{pi} is the output of the neural network. The total error E is also called the total sum of squares error (tss).

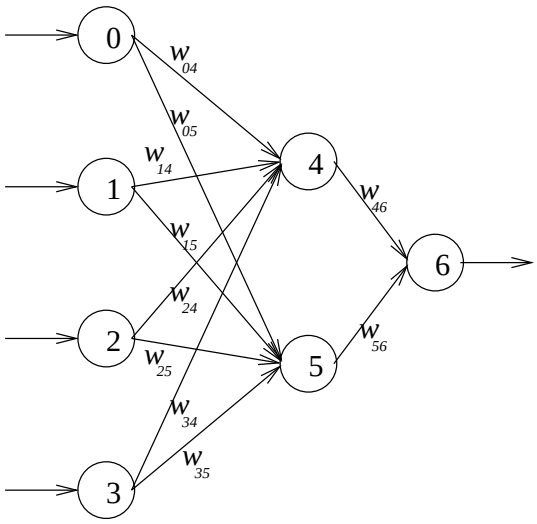


Figure 2.2: An example neural network topology.

The aim in training the neural network is to minimise E. Starting with random numbers assigned to each of the weights, the error is propagated backwards at each epoch through the network by adjusting the weights as follows.

$$\Delta w_{ij} = \epsilon(t_{pi} - o_{pi})a_{pj}$$

for some constant ϵ and a_{pj} the activation of node j for pattern p.

Presenting the entire training set to the network and adjusting the connection weights continues until the error E is a minimum. However, to ensure that the network has generalised the characteristics of the training set the minimum error is taken for an unseen data set. That is, the training data is split into three sets as follows:

training set the cases presented to the neural network for training, the error calculated from the classification of these cases being used to adjust the weights of the network

stopping set the cases presented to the neural network only to be classified, when the error on this set is at a minimum training is stopped

test set the cases presented to the neural network to be classified and are not used at all during training, rather are presented to the trained network to give an unbiased estimate of the error

For further detail of the implementation of perceptron neural networks trained using back-propagation as used here see [138].

2.4.4 Previous Work in Neural Network Classification

The use of neural networks for classification of remotely sensed imagery have been widely investigated, for example see [178, 12, 179, 147, 175, 54]. Many different types of networks have been used more specifically for mapping applications, for examples see [155, 22, 94, 176, 167, 56]. However, only multi-layer perceptron networks trained using back propagation will be considered here as they have been the most widely used and are reasonably well understood.

Xu and Yin [178] trained neural networks for forest management. The attributes used in classification were based on economic return, habitat and soil information. They were able to distinguish, to a limited extent, between areas that should be preserved and others that could be logged. An important aspect of this study is the use of both numeric and non-numeric data. This non-numeric data is not required to have well founded statistical distributions or properties. However, the non-numeric attributes need to be mapped in some way to numeric values to be used in the neural network classification.

Wilkinson et al. [175] reported overall classification accuracy of 92% when using neural networks to map cover types using Landsat TM and SAR data. Classification accuracy was higher when using both types of imagery, over classifications using only one. However, individual forest classes varied in accuracy from 43% to 93%. Again, the problems in vegetation mapping being the quality of the available data and the subjectivity involved in its collection.

Skidmore and Knowles [147] described the use of backpropagation networks for mapping eucalypt forest types[12], using both numeric and non-numeric attributes.

[12]Forest types being scientific groupings of species that are commonly found growing together under particular environmental conditions. The use of forest types is a way of reducing the complexity of forest ecosystems.

While the error rates on the training data were less than 10%, the error on unseen data jumped to between 45–58%. Conventional classification schemes often perform poorly at this type of classification, but in this particular study the main problem was that of inconsistent results. That is, no set of consistent classifications for a given input could be obtained from the networks trained, and this was due to inadequate data. In spite of this, the use of neural networks was still supported due to their ability to identify subtle patterns in the data and model the data.

Milne [113] (published from work carried out as part of this thesis) compared the use of C4.5, neural networks and maximum likelihood classification, using both numeric and non-numeric attributes for binary classification. While there was no statistically significant difference in the accuracy of the classifiers used, the neural network classifier was the preferred solution as a technique was used to reduce the number of false positive or false negative classifications. This technique was developed as part of this work and involves thresholding the outputs of the neural networks. This is discussed further in Section 4.4.

A similar application domain to that of remote sensing is the classification of sounds and speech recognition. Potter et al. [129] used backpropagation neural networks to recognise endnotes of bowhead whale songs. Again, a reason for their use was that neural networks "have been shown to excel at pattern classification where data is noisy and the solution formulation is not well defined". The neural network was able to correctly identify sounds 98.5% of the time. This was a two-fold improvement over the more commonly used spectrogram correlator filter algorithm.

Neural networks are widely promoted as a model free methodology and can be trained to approximate any arbitrary non-linear function. This property makes them a suitable classification technique where there is limited understanding of the domain [71]. Unfortunately, this opinion has meant that pre-processing of data and understanding of the domain has not been sufficiently emphasised in their use. This has lead to inappropriate application of neural networks and disappointing results in some cases. Studies that have included analysis of the domain or pre-processed the available data show better results than those that do not.

Legitimus and Schwab [93] pre-processed data by, firstly, applying a low pass filter to underwater sounds. Free isolated clicks were detected according to the signal

to noise ratio and an energy detection program. Each signal was reduced to 31 attributes using auto regression analysis and Daubechies wavelets analysis. The binary neural networks used produced better results than traditional classifiers (factorial discriminant analysis and clustering), and a multi-class network. Additional improvements of up to 6% on the classification accuracy were achieved for these binary networks using the pre-processed data.

Stretching the contrast of the reflectance values in remote sensing applications can be used to improve classification accuracy [175]. This technique was used throughout this thesis and was also discussed in work published as a result of this research in [113]. As different data types will have different ranges of data values, scaling the data will reduce the dominance of one attribute over another, unfairly biasing the result.

Neural networks have also been shown to be useful when the n-dimensional attribute space that describes each class is not linearly separable. That is, the attribute space of the classes overlap or are made up of interlocking curving boundaries. In particular, this is the reason that neural networks are more appropriate than maximum likelihood classification in some cases. Case studies in the remote sensing literature confirm that where classes are not linearly separable, neural network solutions were better [176, 167, 63].

Neural networks are reasonably fast to generate classifications once trained and can be used in domains for which there is little understanding of the underlying data characteristics. The disadvantage is that their conclusions can not be readily verified or be used to improve our understanding of the domain. This often translates to not enough work being done to optimise their results. A further complication is that while they are reasonably good at handling noisy data, large training datasets are required to mitigate its effects. In this work we look at a number of general techniques that can be used improve the performance of neural networks on small datasets, and that allow automation of the classification.

2.4.5 Decision Tree Classification and C4.5

While it is acknowledged that one form of valuable information in vegetation mapping domains is that of expert knowledge, it is also widely agreed that generating a knowledge-based system from the knowledge of one or more experts is fraught with problems [107, 30]. Ideas need to be communicated by an expert

to a knowledge engineer which then need to be coded in some machine readable form. Problems can arise in communication, consistency and update of rules.

An alternative approach to capturing human knowledge is to learn patterns from a dataset using cases with known class membership. Once the patterns have been identified they can be represented using decision trees. A single decision, or classification, given some set of inputs is a sequence of small decisions that lead to a particular conclusion. An example can be seen in Figure 2.3. Each decision node in the tree gives the test to be carried out and the leaf nodes give the final classification. The decision making path through a decision tree mimics the kinds of rules that human experts generate to reach a decision.

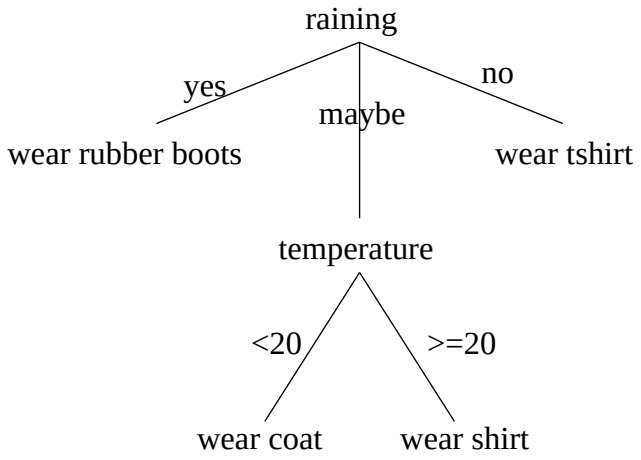


Figure 2.3: Example of a decision tree.

Once the decision tree has been generated from a labelled dataset, it is possible to generate rules from the tree. This makes the model human readable, especially in the case of large decision trees. Rules can be read directly from the tree, for example, for the tree in Figure 2.3 the rules generated are as follows.

```
if (raining = yes) then wear rubber boots
if (raining = maybe) then
      if (temperature < 20) then wear coat
      else wear shirt
if (raining = no) then wear tshirt
```

C4.5 [132] is one of the best known and most widely used classification systems

in the machine learning community. It is a suite of programs that can generate decision trees or a set of rules from a set of training cases.

A decision tree is generated in C4.5 from a set of training cases T, with classes $\{C_1, C_2, ... C_k\}$ in the following way. T is partitioned into subsets $T_1, T_2, ... T_m$ which contain all the cases that have the same outcome for a given test. This is then repeated for each subset until each set contains cases for a single class.

2.4.6 Previous Work in Decision Tree Classification

Decision tree classification is a technique widely used in both the machine learning and remote sensing communities. Its success has been based on the fact that human readable rules are generated, either encoding or extending our understanding of a given domain. An added advantage is that decision tree classification allows the meaningful use of non-numeric data. A sample of work carried out in this area, specifically using C4.5, is given here.

Grigg et al. [67] and Taylor [159] used C4.5 to automatically recognise sounds. Audio data was first processed using a fast Fourier transform to generate a spectrogram, from this a set of attributes was identified for classification. In these cases it was possible to classify at least 80% of the sounds correctly.

Evans et al. [50] investigated the use of C4.5 for predicting areas at risk of salinity. The data used included Landsat TM imagery, slope, aspect and water accumulation information. Mapping areas affected by salinity were able to be identified with accuracies ranging from 61% to 78%. Areas not affected by salinity could be identified with more than 90% accuracy, as these areas made up the bulk of the training data. These results tend to imply that the classifier is not particularly good at predicting areas at risk of salinity. If identifying areas at risk of salinisation was the higher priority the effectiveness of this classifier would not be as high as the overall error rate would imply. So, while C4.5 can handle numeric and non-numeric data it, like all classifiers, suffers from the problems associated with inadequate training data.

Chen et al. [24, 23] used C4.5 classification to recognise lesions in renal biopsy section images. It was possible to automatically recognise boundaries and features within the biopsy sections with 90% accuracy. This work has contributed greatly to the accurate detection of kidney transplant rejections, a task that is typically

done by human experts manually.

Rao et al. [133] used C4.5 for monitoring turf grass from highly subjective data. The classification accuracy obtained was only 59.4%, however, this was a significant increase in the evaluations given by the domain experts. In one aspect of the study the experts refused to give recommendations due to their lack of confidence in their ability to accurately do so.

The use of decision tree classification is widespread due to the simplicity of its application, particularly with tools such as C4.5, and the ease of interpretation of results. In particular, C4.5 has been used for this work due to its ability to incorporate numeric and non-numeric data in a classification.

A specific advantage of using algorithms, such as C4.5, are that they only require a relatively small dataset to learn a given concept [141].

The main disadvantage associated with the use of C4.5 is similar to those of most other classifiers – misclassification errors. In this thesis we look at ways of reducing misclassifications, particularly those due to incomplete or inadequate class information and small noisy datasets.

2.4.7 C4.5 Configuration

A number of parameters can be used to fine tune the behaviour of C4.5.

-s Group discrete values for tests in building the decision tree and each possible value will have a different branch of the tree. Few of the attribute values for this work were discrete values making this flag unnecessary.

-m *weight* Any test used in the tree must have at least 2 outcomes with a minimum number of cases (i.e. at least 2 branches from each node in the decision tree). The default weight is 2.

-c *cf* The amount of pruning to apply to the decision tree. The default value is 25%.

-i *increment* The maximum number of cases that can be added to the window at each iteration. A randomly selected subset of the training cases is used to build an initial decision tree, this is then used to classify the cases not

included in the window, some of the cases that are misclassified are added to the window and the process is repeated until the current window can correctly classify the cases outside of the window. The default value is 20% of the initial window size.

-w *size* The number of cases to be used in the initial window for building the decision tree. The default value of cases from the training set to be used is the maximum of 20% of the training cases and twice the square root of the number of training cases.

-p Soft thresholds use weighting to determine which branch to traverse when the attribute values are close to the thresholds being used, hard thresholds just go down the branch specified by the threshold. The default is for hard thresholds.

Weiss and Hirsh [171] investigated the use of C4.5 to classify noisy datasets. In particular, their aim was to investigate the effect of small training sets on the error rate. The configurations used were:

- the default parameter values with pruning
- the default parameter values without pruning and -m1 (i.e. a decision tree node can be formed be only a single example being covered)

The pruning strategy improved classification accuracy over the use of default parameters in the presence of noise. They found that the small numbers of training cases for each class contributed more to the error rate. When -m20 was used the results were better than the default pruning strategy in the presence of very high (30%) levels of noise. But again, large datasets are required for pruning to be effective.

Baldwin et al [8] used C4.5 to analyse Japanese relative clause constructions. Default parameters were used with 10% pruning, and using 10-fold cross-validation. A base line accuracy of 64.7% was achieved, and was increased to 89% by transforming the data rather than trying to optimise the C4.5 configuration. Similarly Bala et al [7] used C4.5 with the default parameters and used hybrid techniques to improve performance.

Dietterich [41] found that the use of pruning with C4.5 did not provide any statistically significant difference in the in the results. Although this may have been due to the low amount of pruning done.

For the datasets used here a range of C4.5 configurations (i.e. parameter values) were initially tested and no significant improvements in error rate were able to be achieved over just using the default parameter values. However, when the default configuration for C4.5 is used with the techniques described in this thesis the error rates can be reduced. For more detail on C4.5 see [132].

2.4.8 Nearest Neighbour Classification

Another commonly used pattern recognition technique is nearest neighbour classification [35]. It is based on using the training set as a set of prototypes, and does not construct an abstract representation of the data [1]. A classification is given to an unseen case based on its similarity to the prototypes.

When classifying an unseen case a search is carried out to find the prototype case that is most similar. Similarity is measured by the distance to the closest prototype case. Any distance measure can be used, though Euclidean distance is typically used. The distance is calculated attribute by attribute and then summed. The class of the closest prototype case is the class given to the unseen case. The class given need not be that of the single closest prototype case, it can instead be the majority of the k-nearest neighbours, for some constant k [174].

More recently instance-based learning (IBL) was proposed by Aha et al [1] and is a specific implementation of nearest neighbour classification. Three main approaches, IB1, IB2 and IB3, were initially investigated [139]. IB1 stores all training cases as prototypes and the class for an unseen case is the case of the prototype that is closest. IB2 aims to reduce the storage requirements of IB1. This is done by not adding cases to the prototype set when they can be correctly classified by existing prototypes. IB3 extends IB2 by maintaining a record of the number of correct and incorrect classifications for each prototype during training. Those prototypes that result in a large number of incorrect classifications are removed from the prototype set. The discrimination of IBL can also be increased further by considering the classes of the k closest prototypes.

All attributes used in a nearest neighbourhood classification are assumed to be

independent. That is, there is no overlap between the values of attributes that may bias the result. Normalisation of attributes may be of use when the magnitudes of attribute values differ significantly and so reduce bias in the distance to the closest prototype. [174]

Training time is minimal for nearest neighbour classification, however, classification of unseen cases can be expensive because of the need to compare each case with each of the prototype cases. The main advantage of this technique is that it is non-parametric, that is, little is assumed about the characteristics of the dataset [174].

A disadvantage of this classification scheme is that it does not generate a model of the data, and so does not provide any real insight into the general characteristics of the domain. Another disadvantage is, again, that of misclassification due to incomplete class information and small noisy datasets, which we will investigate in this thesis.

Further details on nearest neighbour classification can be found in [35, 44] and for IBL can be found in Aha et al [1].

2.4.9 Previous Work in Nearest Neighbour Classification

Nearest neighbour classification is another technique that has been widely used in both remote sensing and machine learning domains. It is a technique that is easy to implement and it can produce high accuracy results when sufficient training data is available.

Avi-Itzhak et al. [6] proposed a two-phase nearest neighbour classification algorithm for character recognition, a domain similar to classification of remotely sensed images. The two stage approach used, that mapped a character to be recognised first to a broad class then to the actual character, was not only efficient, but also reduced the numbers of misclassifications. In a domain that almost perfect character recognition is being reported, this two-phase approach gave accuracies of over 99%, which was an improvement over other classification packages.

Chittineni [26] describe the use of nearest neighbour classification from data that contains errors, as can be found when using remotely sensed data. Techniques were given for error correction using the probability of errors occurring.

Breuel [17] used nearest neighbour classification to recognise hand printed digits from noisy data. Post-processing using decision trees was also carried out and improved the classification accuracy. Accuracies of up to 99% were reported for the combined technique.

Ince [74] compared nearest neighbour and maximum likelihood classification. Landsat TM data was used to map seven classes including wheat, fallow land, roads and villages. Accuracies of around 90% were reported for both classifiers and it was found that nearest neighbour classification was more accurate and robust than maximum likelihood classification.

2.4.9.1 Nearest Neighbour Configuration

Nearest neighbour algorithms are useful for recognising cases that are in some way similar to cases that we know the class membership of, however, they can be a poor classification scheme for the following reasons.

- All attributes are assumed to be effectively independent, that is there is no overlap between the values of attributes that may bias the result.
- The variance of attribute values has been normalised.
- They can exhibit poor generalisation as they provide essentially a specific model of the training data.
- Classification times depend on the number of cases in the training set.

IBL has been used in this work using the IB1 approach of storing all cases in the training dataset as prototypes. It was chosen because some of the disadvantages of this technique are offset by the classification framework used. Classification times will remain within reasonable bounds as in this thesis we are only interested in domains with small training datasets. Specifically, however, we are interested in the property of poor generalisation – a reasonably specific model of the training data is exactly what we want in the context of this work.

2.4.10 Unsupervised Classification Techniques

Typically, reliable classification requires large amounts of training data which, as already stated, is a problem in domains using remotely sensed data. If we pre-process the available data we can simplify large amounts of information to highlight particular details within the data or remove noise from the data.

Unsupervised classification, or clustering, has the advantage of grouping instances with similar properties and so simplifies the data [81]. Rather than forcing a set of classes onto the data set, an unsupervised classification scheme will reflect natural clusters in the data. This can be useful for finding novel or interesting features in the data that would not otherwise be found. It can also be used to highlight significant information by reducing the dimensionality of the data.

One unsupervised classification technique used in remote sensing domains is ISOCLASS. It is essentially the iterative optimisation algorithm of Ball and Hall [9, 125]. A set of points is chosen in multidimensional space that serve as the centre value of each cluster, or class. Training cases are moved from one cluster to another, to minimise the Euclidean distance to the centre of each cluster. The centre point of each cluster, which is the mean of all points in the cluster, is recalculated for each iteration. [134]

AutoClass [21, 69, 168] is another tool used for data exploration and knowledge discovery. It is not intended as a one off classification algorithm, rather as a tool to enhance the work of an expert in classifying data. It has successfully been applied to many different domains, including classification of remotely sensed imagery [64, 80]. Further discussion of this algorithm is given in the following section.

A classification generated from a clustering algorithm is typically mapped directly to real world classes, or investigated by experts to improve their understanding of the given domain. An alternative is to use the classification as an additional attribute in other classification tasks. By using unsupervised classifications in this way we can utilise their ability to highlight relevant information in the available data. Putting this another way, we can use these techniques to help reduce the irrelevant information in any available data and so make the most of the information contained in it.

2.4.11 AutoClass Classification

AutoClass [21, 69] is well known within the data mining community as a data exploration and knowledge discovery tool. Although it essentially generates an unsupervised classification, it is not intended as a one-off classification algorithm, rather as a tool to enhance the work of an expert working with large and complex datasets.

AutoClass is an unsupervised Bayesian classifier that searches for the model that best describes a given dataset. As part of this, it determines the appropriate number of classes as well as the level of complexity required for each class. The most probable classification given the data is such that the members of a class are most predictive of each other, giving a domain independent measure of similarity [64]. This is of particular use for exploratory data analysis. The data presented to AutoClass can be a mixture of discrete or real values and there can be correlations between attributes within a class. It can also handle missing attribute values, although all attributes are assumed to be relevant.

Processing time is approximately linear in the amount of data. This means that large amounts of data can be classified using AutoClass reasonably efficiently.

Rather than a simple partitioning of the space, AutoClass tries to find the best class description. For a given data set, the aim is to find the most likely probability density function, and find the maximum posterior parameter values for a given probability density function. An outline of the algorithm is as follows.

Given hypothesis H and evidence E we define

$\pi(H)$ the prior probability, that is, the belief in H prior to or in the absence of evidence E

$\pi(H \mid E)$ the posterior probability, that is, the belief in H after observing E

$L(E \mid H)$ the likelihood of each possible evidence combination E in each possible world H

It is assumed that the world is in some state and that some evidence will be observed, and so, $\sum_H \pi(H) = 1$ and $\sum_H L(E \mid H) = 1$.

The joint probability J of E and H is

$$J(E \mid H) \equiv L(E \mid H)\pi(H)$$

and the normalised joint probability (Bayes rule)

$$J(E \mid H) \equiv \frac{L(E|H)\pi(H)}{\sum_H L(E|H)\pi(H)}$$

In theory it is possible for a given situation to choose a set of states H, an associated likelihood function describing what evidence is expected to be observed in those states, a set of prior probabilities on those states and to collect relevant evidence. However, in practice these sums will be intractable.

So rather than considering all possible states H, we assume the model falls into some smaller space S, and refine the model. The parameters that can describe the model are

1. A probability density function T that describes the general form of the model.
2. The free variables in the general form V.

Given T, V and S, the likelihood function becomes, $L(E \mid VTS)$ and the prior probability $\pi(VT \mid S)$. If we also have that H and E are continuous then the joint probability becomes

$$dJ(EVT \mid S) = L(E \mid VTS)d\pi(VT \mid S)$$

which typically cannot be normalised. Combinations of RT are searched until the marginal joint probability

$$M(ERT \mid S) \equiv \int_{V \in R} dJ(EVT \mid S)$$

is as large as possible. Searching for better models of RT can be stopped when estimates of how long it would take to find a better model become too large.

Further detail on the AutoClass algorithm can be found in [168, 21, 69].

2.4.12 Previous Work using AutoClass

The use of AutoClass was demonstrated by using it to reclassify the Infrared Astronomical Satellite (IRAS) Low Resolution Spectra (LRS) Atlas [64]. Previously, the more than 5000 stellar infrared spectra in the atlas, were organised according to spectral features. However, this was found to be an inappropriate representation in some cases. AutoClass was used to reclassify the spectra automatically, and found the previously known classes as well as some new ones. AutoClass was useful for finding subtle differences in spectral signatures, allowing differentiation that would otherwise have been impossible. It also allows large amounts of data to be explored and classified reliably, rather than having to analyse it manually.

Kanefsky et al. [80] used AutoClass to investigate classes in a Landsat TM image for an area in Kansas, U.S.A., containing only crop and grazing land, and marginal woodlands. AutoClass identified 93 classes in the image, the majority of which were able to be mapped directly to meaningful physical features.

Due to the success of AutoClass in highlighting information it was chosen as for use in this work. Further discussion of this can be found in Chapter 5.

2.5 Accuracy Assessment

Error rates of 20-40% are not uncommon when classifying remotely sensed data, and for specific classification tasks higher error rates have been reported. Even if high error rates are considered acceptable in a given domain we still have the problem of how reliable these estimates really are.

Higher resolution data or aerial photographs have been used to provide training data for lower resolution data. Ripple [135], for example, used colour aerial photographs to determine the accuracy of a canopy cover classification from AVHRR data. This kind of assessment assumes that accurate information can be derived from the higher resolution data by a human interpreter.

The classifications provided by photo-interpretation have also been used for assessing the accuracy of classification of other lower resolution remotely sensed data. While it is accepted that there will be some inconsistency or errors in such classifications they are assumed to be correct. This can lead to invalid results and

unfair assessment of the classifications generated as the true accuracy of the lower resolution classification is difficult to determine.

Until recently the same data was often used to train and test classification systems in remote sensing and vegetation mapping domains [33, 105]. This can lead to invalid measures of accuracy as the classifier may not give a generalised representation of the concepts within the data. Ideally an independent dataset, that has not been used in training, should be used to test the performance of the classifier and determine its error rate. The absence of a test dataset makes it impossible to give a reasonable estimate of the true error rates on a classifier.

Other factors affecting the accuracy of a classification that must be taken into account when using remotely sensed data are the methods for collecting ground truth data, the classification scheme being used, spatial autocorrelation[13], sample size and sampling scheme [33]. The accuracy of the ground truth data should be known, although in practice this is difficult and it is assumed to be correct.

The sample size should be large enough to contain statistically significant numbers of cases in each of the desired categories. Unfortunately, the sample size must be balanced with the cost of collecting the data. Richards [134], states that for n attributes being used in a classification $n+1$ training cases are required to avoid the covariance matrix being singular. A rule of thumb, suggested by Congalton [33], is 50 samples for each class in the classification. Swain and Davis [156] recommend a minimum of $10n$, and if possible $100n$, training cases should be available. However, in practice these amounts of data are not necessarily going to be possible.

The sampling scheme determines which sites are to be surveyed for the given study area. Poor choice of sampling scheme can seriously bias the occurrence of particular classes in the classification. For example, just surveying vegetation along a valley would give very different species to surveying along ridges.

Ideally we would have large amounts of accurately classified data, that has been generated with an appropriate sampling scheme, for use in training classifiers. Where such data are available we are able to produce classifiers that have accurate representations of the domain and produce low error rate classifications when presented with new data. Unfortunately in vegetation mapping domains this is rarely possible. The central theme of this thesis is to address the problems

[13]Spatial autocorrelation is where a pixels class is not independent of the class of the neighbouring pixels.

associated with limited availability of high quality training data, and aims to produce the best possible classifications from the data that is available.

Once we have some data available for training classifiers it should be partitioned into training and test sets, and for this there are a number of approaches that can be taken.

The hold-out method [85] uses two thirds of the data for training the classifier and keeps the remaining third for testing the classifier. The test set is not used at all during the training of the classifier. The training and test sets can be found by random sub-sampling which can be repeated a number of times to estimate the standard deviation of the accuracy assessments. The holdout approach is used through out this work.

Another approach, similar to random sub-sampling, is n-fold cross validation [85]. It has been used widely within in machine learning community to improve the estimate of classification accuracy. Examples of its use can be found in [85, 173, 45, 172, 16]. It involves splitting the available training data into n mutually exclusive subsets of approximately equal size. The classifier is trained n times on all of the data in $n-1$ of the partitions and tested on the remaining partition. The error estimate is the average error over each of the n test partitions.

If the accuracy estimate is highly variable over each of the test partitions the error estimate provided by sub-sampling techniques is likely to be unreliable [85]. This is generally the case in vegetation mapping domains, that is error rates across partitions can vary significantly. The classification framework presented in this thesis has been developed to address such consistency problems. In addition to the consistency problems these techniques are a computationally expensive method of determining classifier accuracy. For these reasons cross-validation is not used in this work. This is discussed further in Chapter 8.

Finally the error rate needs to be presented in a meaningful way. The simplest and most common method for representing the accuracy of a classification is the confusion, or error, matrix [28]. Using such a matrix you can obtain the error rates on individual classes as well as the overall error for a classifier. This is the means by which error rates will be discussed here.

2.6 Conclusions

While the classification of remotely sensed data has been extensively investigated there is still more work to be done. In this thesis we look at a number of techniques for improving and automating the classification of remotely sensed data using existing classification algorithms.

In this Chapter we introduced a broad range of topics across a number of research areas. Each of these will be discussed further and built on throughout the thesis. In particular, to aid readability, further discussion of the literature in specific areas has been included in the relevant sections as required.

Chapter 3

Overview of the Image Datasets Investigated

Two study areas were used for this work and are described here.

3.1 Charles Sturt University

A single airborne video image was acquired using the airborne video system (ABVS) developed at Charles Sturt University [87]. An ABVS image[1] contains 737×537 pixels and for each pixel four spectral values are measured.

The ABVS image was acquired over the Charles Sturt University campus (*CSU*) in Wagga Wagga, NSW. It contains open forest, water and urban areas as shown in Figure 3.1.

While majority of the image contains vegetation, particular features of interest have been labelled and are as follows.

B Buildings with landscaped gardens surrounding them. Many trees surround the buildings, some trees overhanging the roofs. Many areas of green lawn are also maintained around the buildings.

[1]Each remotely sensed image is essentially n images, one for each spectral band. In the case of an ABVS image there are actually four independant sensors that measure data in a given spectral range, while for the Landsat TM images each image is generated by splitting the reflectance measured by a single sensor.

Figure 3.1: Identification of areas of interest for the *CSU* image.

C Car parks with a bitumen surface.

D Dirt tracks.

F Cultivated field.

G Open forest that is fenced off for grazing. This area is bounded above and below by roads and to the right by a fence - showing as a lighter colour background. The soil is far more likely to show through here as there will be less vegetation cover.

L Lakes or dams. The spectral signatures for these are likely to be confused due to trees along the banks that overhang the water, water may be muddy or covered with water weeds. In particular, the fourth lake from the right was covered with azolla (a water weed) that is red when in sunlight (as is the case here) and green when in shadow.

O Open forest - a mixture of eucalypts and conifers. A hill extends, sloping downwards to the area labelled G, the remainder of the landscape being much flatter. The under-story consists of grasses and weeds.

P Landscaped trees with low cut grass and some weeds underneath. Some areas, particularly to the right of the image, are watered and so are much like lawns.

W Water storage tanks on the top of the hill.

A colour composite of the image can be seen in Figure 3.2.

Figure 3.2: Colour composite of the *CSU* ABVS image.

3.2 Royal National Park

Royal National Park, in NSW, Australia, situated approximately 30km south of Sydney. Although it is the second oldest national park in the world, being declared a national park in 1879, until recently a vegetation map of the area was not available. Automatically generated vegetation maps would be a particular advantage in this case for a number of reasons. Due to the proximity of the city the impact of human activity can be significant and so should be monitored. Detailed surveys would be difficult to carry out on a regular basis, even for an area of this size and accessibility.

The park is bounded roughly by Port Hacking and suburbs to the north, the Pacific ocean to the east and limited suburban areas to the west and south as shown in Figure 3.3.

Figure 3.3: Royal National Park.

The area investigated was Audley (identified here as *RNP*) as it is easily identified

and has a range of cover types (urban, river, landscaped and bush areas). The Audley area can be seen in Figure 3.4.

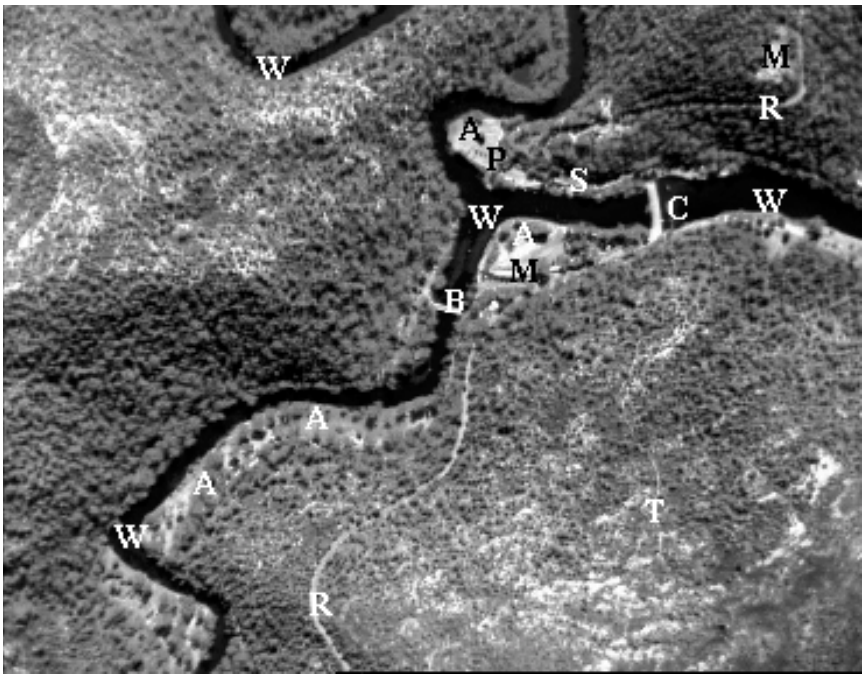


Figure 3.4: Identification of areas of interest for the *RNP* image.

Particular features of interest have been labelled and are as follows.

A Picnic areas. Maintained lawns and a number of exotic tree species.

B Wooden bridge.

C Cement causeway.

M Buildings.

P Car parks.

R Bitumen roads.

S Boat shed with a wooden platform that extends to the waters edge.

T Walking tracks.

W Fresh water river. In places quite shallow with water weeds.

Sensor	Path/Row	Date	Resolution	Corrections
ABVS	N/A	10/01/97	$2 \times 2m$	none
Landsat TM	089/084	30/05/96	$30 \times 30m$	geocoded
Landsat TM	089/084	02/08/96	$30 \times 30m$	geocoded
Landsat TM	089/084	08/12/96	$30 \times 30m$	geocoded

Table 3.1: Acquisition information for the *RNP* dataset.

For this dataset two types of remotely sensed data were available – a single ABVS image and three Landsat TM images. Acquisition details for the images can be seen in see Table 3.1.

Information about data corrections can be found in [28, 134].

A colour composite of the ABVS image can be seen in Figure 3.5.

Figure 3.5: Colour composite of the *RNP* ABVS image.

3.3 Generating Training Data

The term dataset in the context of this work will typically mean the attributes available for a particular area. Attributes could potentially be a variety of remotely sensed data, as well as climate, soil or other information. However, in this work we will use only remotely sensed data. Each case in the dataset corresponds to a pixel[2] in the image, and each will have a value for each of the attributes. Further discussion of attributes used for classification can be seen in Chapter 5.

Each dataset will have a sub-set of cases which is the training dataset, where the class of each case is known. The training data is divided into three further sets, a training set, stopping set and test set. The training set contains the cases presented to a classification system for training. The test set contains the cases presented to the classifier to give an estimate of the error. The test set is not used at all during the training of the classifier. The stopping set is used for training the neural networks only and is used to determine when to stop training, and will be discussed further in Chapter 4.

When surveying a study area to generate a training dataset random selection of sites may be done. The training datasets used here were generated as follows. Firstly, areas for which class labels could be reasonably accurately determined were manually identified from an image. Then, a random selection of n cases for each class is made from these areas. Congalton [33] suggests that around 50 cases were generated for each class in a given classification task attributes. This means that for a single image we may have at most 1% of the pixels in an image for training. This, however, is not unusual in this domain.

Generating accurate training datasets is a significant problem in itself and can not possibly be addressed thoroughly here. Errors in training datasets can arise for a number of reasons, including incorrectly classified cases, noise in the data due to incorrect pre-processing of the images and incorrectly locating survey sites within an image. However, the aim of this work is, in part, to reduce the misclassifications due to the inaccuracy of generating training data.

[2]The terms case and pixel may be used interchangeably and the meaning should be clear from the context.

Chapter 4

Issues in Neural Network Classification

Neural networks have been promoted as the answer to all pattern recognition problems since their introduction in the 60's. While extremely useful in pattern recognition problems, their ability to produce meaningful classifications from small noisy training datasets has been greatly exaggerated. However, as we shall demonstrate, it is possible to obtain useful classifications from such data using neural networks. In this chapter we introduce the types of networks that will be used throughout this thesis and outline some of the methods by which network performance on small noisy datasets can be improved and automated. The concepts introduced here will be discussed later in more detail in the relevant sections.

4.1 Determining Network Topology

Determining the best topology for a neural network for a specific classification task is a problem that is largely done by brute force search. The main problem is that different configurations of a network can result in wildly varying results. Gahegan et al [61] state that one of the reasons there has not been wider acceptance of neural networks in GIS and remote sensing domains is the difficulty in configuring a network.

Determining the number of input nodes for a network is generally easy – one

input node is required for each attribute in the dataset. Similarly, the number of output nodes is going to be constrained in some way by the number of classes to be identified. Determining the remainder of the configuration is a little harder.

The number of hidden layers and number of nodes in each of these layers also needs to be determined. The aim is to maximise the performance of the network is maximised. Determining the topology the hidden layers presents the greatest problem in configuring a neural network for a given classification task.

Rumelhart [138] suggested that networks with more hidden layers, and fewer nodes in the earlier layers may generalise better than those with few layers and more nodes in those layers. However, networks with many layers are harder to train [143].

Much of the work reported in the literature use a single hidden layer as there is little in the way of compelling evidence to support the idea that more than one hidden layer improves the performance of a neural network [129]. However, this does not mean that the use of multiple hidden layer networks can be completely discounted. It has been shown that problems that involve an exclusive-or type operation can not be solved with a single layer network [143]. An explanation that has been given for using multiple layers is that the first hidden layer may be extracting features of the classes that can be interpreted by later layers [143].

Nikolopoulos and Fellrath [119] used a neural network to detect interest rate trends. A four layer network with 33 inputs, three nodes in the second layer, two nodes in the third layer and one output node was used. This configuration was found by using a genetic algorithm based system for configuring neural networks. The accuracy reported on this network was 71%, and presumably was the highest of all configurations trialled. Only a small amount of training data was available[1] for this work, indicating that multiple hidden layers may be appropriate in this situation.

Skidmore and Knowles [147] found that when classifying forest types, three hidden layers gave better results than those with only one or two layers. The dataset used for this work was small and noisy, and contained numeric and non-numeric attributes. Again, this supports the idea that multiple hidden layers may provide an advantage for small noisy datasets.

[1] 75 cases for training and 25 cases in the test set.

Jarvis and Stuart [75] gave a summary of network topologies used in remote sensing literature and found that around 25% used more than one hidden layer.

Overall there is little agreement about how many hidden layers there should be for neural networks. The literature seems to indicate that we can not even narrow down a set of rules for determining an appropriate number of hidden layers for classifications tasks within a single domain.

The second issue is determining the number of nodes in each of the hidden layers. It has been suggested that the number of hidden layer nodes needs to be at least the number of classes in the given classification task [129]. A common practice is to use the geometric or arithmetic mean of the number of input nodes and output nodes for the hidden layer [129].

Sietsma and Dow [143] found that networks with the minimum number of nodes in the hidden layer did not generalise as well as those with a "larger" number of nodes. It has also been found that extra nodes in the hidden layer can help to remove local minima which contributes to the generalisation ability of the network [138].

Gahegan et al [61] and German and Gahegan [63] discussed ways of improving classification accuracy for neural networks. In particular, a rule for choosing the number of hidden nodes in single hidden layer networks was given – $\binom{n}{2}$, where n is the number of classes. The reason for choosing this number is that it is the number of pairwise discriminant functions needed to separate n classes, and each node should behave as one such function.

The results for neural networks trained using $\binom{n}{2}$ hidden nodes gave similar results to that of a maximum likelihood classifier. No comparison was made with alternative network configurations. Most importantly, however, this work demonstrated that if a reasonable network configuration can be found the performance can be tuned using other methods.

Skidmore et al [146] investigated varying the number of hidden nodes in a neural network for mapping forest types. Each of the 14 attributes were mapped to 14 input nodes, and each of the 5 classes were represented by a single output node. A single hidden layer was used and the number of nodes in this layer was varied from one to 20. The test set error varied quite significantly for the different topologies, from 45% to 95%. The optimal number of hidden layer nodes was found to be 10. Adding more hidden layer nodes only served to reduce the generalisation

ability, supporting the use of the rule given by Gahegan et al. [61]. However, the dataset used was small and noisy, which may also account for the wildly varying accuracies.

Potter et al [129] investigated the use of neural networks for identifying whale song endnotes from other sounds. This was a two class problem, giving classifications of a yes or no answer for the detection of whale song end notes from spectrograms. Experimentation with network topologies found that the best configuration was four hidden nodes for 192 inputs and one output node. The Gahegan et al. rule [61] would suggest only one hidden node.

Battiti and Colla [10] used neural networks for character recognition, specifically to recognise digits from image data. A number of different networks were trained, each using different attributes extracted from the image data. Five networks were trained with the configurations shown in Table 4.1. In this case, the Gahegan et al rule [61] would suggest using 45 hidden layer nodes. However, as can be seen in Table 4.1 all networks gave error rates that were similar. Even for networks that had very different numbers of hidden layer nodes, from those suggested by the Gahegan rule, the error rates were low.

# Inputs	# Hidden nodes	# Outputs	% Test Error
28	28	10	5.29%
48	28	10	5.4%
32	64	10	6.83%
56	32	10	5.03%
45	45	10	5.32%

Table 4.1: Network configurations and error rates reported in [10].

Jarvis and Stuart [75] found from their surveys of the remote sensing literature that the number of hidden nodes is typically more than the number of nodes in the input layer. They suggest that a larger number of hidden nodes may be required to classify remotely sensed data, particularly scenes with greater complexity and granularity.

Sigillito and Hutton [144] suggest that for radar signalling applications the number of nodes should be less than half the number of inputs. Blum [14] narrows this down further by saying the number of hidden nodes should be between the number of input nodes and the number of output nodes.

Overall, too few nodes in the hidden layers and the network may not be able to distinguish between each of the classes. Too many nodes and the network may not converge in a reasonable time, and the network generalisation decreases. The only generally agreed upon guidelines given are that one hidden layer is usually sufficient and the number of hidden nodes should not be "too large" or "too small".

Sietsma and Dow [143] took an alternative approach to configuring networks, called pruning. A three layer network topology is chosen that is larger than the anticipated minimum requirements, the initial configuration being based on intuition or experimentation. After the network is trained the nodes that are not contributing to the solution (stage one pruning), as well as the nodes that are not contributing to the next layer (stage two pruning) are removed. The non-contributing nodes are those that have approximately constant outputs across the training set or have outputs that mimic the outputs of other nodes. The nodes that are not contributing to the next layer are determined by a minimum information content criterion. To ensure linear separability is maintained after stage two pruning, additional layers can be added to the network. The results of this work showed that stage two pruning, and the subsequent addition of hidden layers where required, did not provide robust performance, and that stage one pruning was the significant step. Most importantly in the context of this work, it was also found that stage one pruning did not substantially improve network performance in the presence of noise. This makes pruning inappropriate for the datasets being used in this thesis.

In contrast to the majority of discussions on network topologies in the literature some authors report that topology is not necessarily a defining factor in classification accuracy. This idea is central to the work in this thesis.

Jarvis and Stuart [75] carried out experiments on network topologies to classify land cover types of water, built and vegetation from Landsat TM data. Three layer networks were trialled with six inputs[2], three output nodes and varying the number of hidden nodes from three to 15. They concluded that these networks were insensitive to the number of hidden nodes as the classification accuracies did not change significantly. However, they also state that the robustness of the networks trained may be due to the large amounts of data used in training.

Rogova [136] trained a number of networks with varying numbers of hidden layers

[2]TM band 6 was excluded from the set of input attributes.

and nodes in the hidden layers for character recognition. In addition to this, networks were trained on different sets of attributes. The results of the individual networks were then combined using the Dempster-Shafer theory of evidence. The results were better for the combined networks over individual networks. But more importantly, in the context of this work, combinations involving the use of different attributes in training performed better than those for which different network architectures were used. That is, other factors can mitigate the effects of a less than optimal network topology. Again, an idea that is central to this thesis.

Lees [92] states that most emphasis on neural network classification is placed on the algorithm and not enough emphasis is given to analysis of the available data. He points out that no matter how sophisticated an algorithm is inadequate data will result in poor results.

In this thesis the focus is shifted towards investigation of the data and making the most of the available information. We do not focus on finding an optimal topology for the networks used. Here we find a workable configuration that allows automation of neural network classification and use other techniques to fine tune the results.

4.2 Network Configurations Used

The guidelines discussed in the previous section were used initially to help determine network topology. That is, it is generally agreed that a single hidden layer is sufficient and that performance of a neural network is degraded if the number of nodes in the hidden layer is too small or too large.

The work of Gahegan et al [61], German and Gahegan [63], Jarvis and Stuart [75] and Rogova [136] demonstrate an important point, that the topology of an individual network is not always the single most critical factor in maximising performance. These papers demonstrate the use of techniques used to improve overall classification accuracy, in addition to the choice of a "reasonable" network topology. In the same way here, a reasonable network topology was determined, via extensive experimentation, and is used in conjunction with other techniques to improve classification accuracy. The techniques used to achieve this will be discussed in later chapters.

The aim of a neural network, in the context of this work, is to reduce a large number of overlapping and interacting, noisy attributes into a single concept (i.e. a single target class or classification). That is, the classification task is broken down into a number of simple tasks, rather than training one large classifier to recognise everything. The simpler the classification task the easier it is to recognise the general characteristics of a given class from a small noisy dataset. We can then combine the results of the simple classifiers. Specifically classification is simplified in the following ways.

Binary Classification Each classifier should only be trying to distinguish between two classifications. By doing this we reduce the number of attributes required and are able to improve the classification accuracy of individual classifiers [76].

Hierarchical Classification Classification should be hierarchical in nature. That is, start with classifying an image into broad classes, such as vegetation and water. Then each class can be further broken down into its sub-classes, for example segmenting a vegetation class into grass and trees, or forest types.

Experiments with a range of neural network topologies, varying the numbers of layers, number of hidden nodes in each layer, and number of output nodes were trialled for small noisy datasets. The overall accuracy was often consistent over all network topologies trained. A significant difference in performance was, however, provided by the use of a binary structure in the classification task. More consistent classifications were obtained for individual test cases, across multiple networks, by training each network to determine class membership for a single class only.

As we shall see, throughout this work, single output binary classifiers are preferable to networks with one output node for each class as they allow us to better deal with noisy data and class separability issues. Experiments with networks with more than one output, on small noisy datasets, resulted in output values all clustering around the same values. This meant that class membership is not easily determined and so high error rates result.

As we shall see in later chapters a single output binary network provides more accurate and consistent results. This approach also provides a way for us to explicitly handle misclassifications. Thus, unless otherwise stated, the neural networks used in this thesis have the topologies as given in Figure 4.1.

- A three layer network is used – one input layer, one hidden layer and one output layer.
- There is one input node for each attribute in the dataset.
- A single output is used to give a binary classification.
- The number of hidden nodes is half the number of input nodes, if the number of input nodes is less than or equal to 40. If the number input nodes is greater than 40 the number of hidden nodes defaults to 20.
- The network is fully connected, that is, each node is connected to all nodes in the previous and following layers.
- A learning rate (ϵ) of 0.4 was found by experimentation to be the most appropriate value for the datasets investigated here. This value reduced the incidence of oscillation in the total sum of squares error and seemed to reduced the possibility of finding a local minimum.

Figure 4.1: Neural network configurations used throughout this thesis.

4.3 Input and Output Values

Neural networks will perform better when the input values are all of the same order of magnitude. For this reason all values presented to the input layer of the neural network were scaled to be between 0 and 1. This was achieved by simply dividing each number by the maximum value for that attribute.

Target classes were mapped to output values between 0 and 1, exclusive. This is because training perceptron networks with target values of exactly 0 or 1 can result in problems with the backpropagation of the error. In particular the single output binary networks that were used to give a yes/no answer for class membership of a single class, used target classes of 0.1 for no and 0.9 for yes.

4.4 Thresholding Output Values

Neural networks produce continuous output values in a given range, which for this work was values between 0 and 1. These output values need to be mapped to target values in a meaningful way. Typically, output values are divided into

a number of sub-ranges, which are mapped to the values representing the target classes.

For example, if we have a four class classification problem we may map each class to one of the binary vectors in the set [0, 0], [0, 1], [1, 0] or [1, 1]. This would require two output nodes to represent, both of which would produce values ranging from 0 to 1. To avoid problems with the backpropagation of the error during training, the target output values can be represented by the vectors [0.1, 0.1], [0.1, 0.9], [0.9, 0.1] and [0.9, 0.9] respectively. The output values from each node can then be given two ranges - 0 to 0.5, and 0.5 to 1. If the value from an output node is in the range 0 to 0.5 it has an output class of 0, and if the range is 0.5 to 1 it has an output class of 1. The mappings for these outputs can be seen in Figure 4.2.

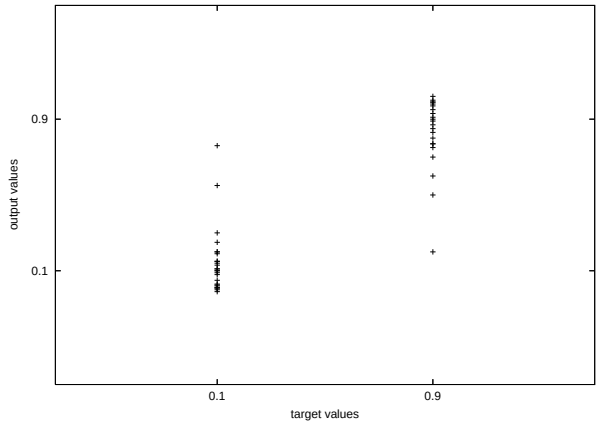


Figure 4.2: Typical class separation for the outputs of a neural network.

Early work with small noisy datasets showed that simply dividing the output values for each node into equal ranges resulted in much higher error rates. The output values for these datasets tend not to cluster around target values. It was found that clusters were less clearly defined and often closer together, as shown in Figure 4.3. This however is not necessarily a limitation as smaller ranges for the output values being mapped back to the target values can be used to ensure that the majority of pixels are still classified with a reasonable level of accuracy.

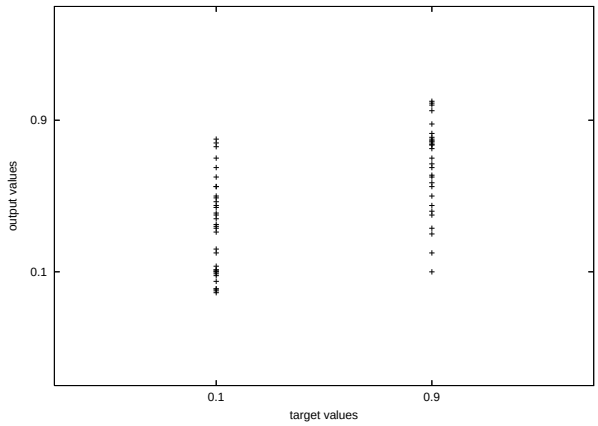


Figure 4.3: Outputs for a neural network trained on small, noisy datasets where class separation is poor.

Alternatively, the threshold between classes need not be an equal split of the output value range. Extensive experimentation showed that, with the binary networks as described in Figure 4.1, rather than using a fixed threshold to give fixed ranges determining class membership, a varying threshold could be used. That is, in the case of the binary networks, if the threshold of 0.7 was used, an output value between 0 and 0.7 would be mapped to the target value 0.1, and an output value of 0.7 to 1 would be mapped to 0.9. The actual threshold used can be chosen to minimise the number of false positives and false negatives.

The concept of thresholding neural network outputs was also reported in [113]. A comparison of the classifications given by C4.5, maximum likelihood and thresholded neural networks to distinguish high level tree species was carried out. It was found that the neural network results could be refined using thresholding to give the minimum number of misclassifications when compared with the other techniques. That is, by choosing an appropriate threshold we are able to reduce the number of pixels that are classified as being in a given class when they are not.

Varying the thresholds for determining class membership from the output values works for a number of reasons:

- It is not easy to obtain clear class separation for small training datasets without specialising the neural network to recognise the training data only. i.e. the network is unable to generalise the information it has learnt and so

is unable to classify unseen data with a reasonable degree of accuracy.

- Insufficient class separation in a dataset will cause the outputs of the neural network to spread across the range 0.1 to 0.9, rather than clustering closely around the target output values.

Thresholding the output values can be done in one of two ways:

- A case is only given a classification when its output value is within $\pm e$, for some e, of a target value.
- A variable threshold is used that splits the output values into two ranges that are not equal in size.

The second method is used most in this work as we are training networks to determine a yes or no answer for membership of a single class. A threshold is chosen that minimises the number of false positives and false negatives. The result of this is that unless the outputs cluster reasonably close to the target yes output value of 0.9, the classification given will be no.

It is still possible for thresholding to be ineffective in generating meaningful classifications if all outputs cluster within a small range of values, and show no real separation at all. This tends to happen when the neural network has been swamped with too many classes or too much information. The problem is compounded if the data is noisy or irrelevant. In this case a default classification occurs, that is, all outputs are essentially the same value which would all map to the same target class.

The use of thresholding neural network outputs will be demonstrated in Chapter 8. The application of thresholding was also published in [113], a link for which can be found in Appendix B.

4.5 Determining When to Stop Training

As discussed in Section 2.4.3 training a neural network stops at the point at which the error on the stopping set is at a minimum, indicating that the further training of neural network would begin to specialise to the training data, rather

than generalise the information. It is quite simple to plot the value of the total sum of squares error, or tss, during training for the training and stopping sets, and then use that to determine visually the point at which training should stop.

A neural network is trained iteratively by presenting each case in the training dataset to it for classification. In each iteration of training, each case is classified by the neural network and the difference, or error, between the target value and the actual output value is determined. The error is then used to adjust the weights and the training dataset is presented to the neural network again. Each iteration of presenting the dataset to the neural network and then adjusting the weights is called an epoch.

The point at which to stop training a neural network is the epoch at which the total sum of squares error (tss) on the stopping set is at a minimum:

$$tss = \sum_{p} \sum_{i} (t_{pi} - o_{pi})^2$$

where p ranges over the set of input patterns and i over the number of outputs, t_{pi} is the target output and o_{pi} is the output of the neural network [138].

The tss is a measure of the closeness of the approximation to the optimal solution and will generally decrease over the course of training the neural network [138].

Stopping the training of a neural network too early results in a meaningless classifier with none of the general characteristics of the training set. The classification given by the network is largely determined by the initial random weights when training is stopped too early. It is also widely accepted that a network is still not trained while the tss is oscillating between high and low values.

Through extensive experimentation the author found that for this type of data a stopping point less than 50 epochs indicates that the network has not had a chance to learn anything. The choice of this threshold is demonstrated by the graph in Figure 4.4, showing a typical plot of the tss at each epoch of training. As can be seen the tss is still oscillating within the first 50 epochs and the minimum error occurs here, at around 45 epochs. Choosing this first minimum as the stopping point results in poor generalisation and inconsistent results. If we choose the true minimum, at around 260 epochs, the tss has stabilised and the error on the test

set is at "a minimum". For the types of networks and data used here the second minimum error on the stopping set has been found to be a more appropriate choice.

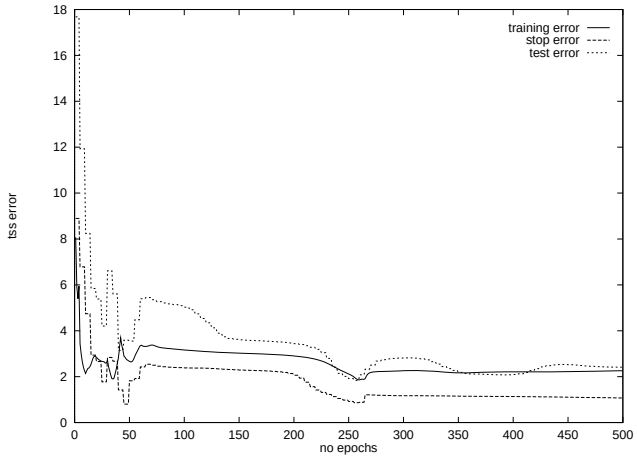


Figure 4.4: Total sum of squares error on training, stopping and test datasets.

Conversely it was also found that if a neural network is trained for more than 500 epochs, with the given topologies on small noisy datasets, it is usually the the case that network has over-fitted the data. That is, it has learnt the specific details of the training set and is unable to generalise the characteristics of the classes for use with unseen data with reasonable accuracy. German and Gahegan [63] similarly found that more than 1000 epochs was unnecessary.

Determining when to stop training is a little harder to achieve automatically, but has been done for the neural networks in this thesis using the following heuristics. These heuristics were derived as a result of extensive experimentation carried out over a period of time on small noisy datasets.

1. Train a neural network for 500 epochs – as there are such a small number of training cases the networks were found to have specialised the information in the data, rather than generalising it, if training continues past this point. The sum of squares error on the stopping set at each epoch is recorded by classifying the data, but, this data is not used to train the network.

2. The epoch at which the minimum tss (i.e. the minimum error) occurs for the stopping set becomes the stopping point in the training of the neural network. The point at which the tss at epoch $t + 1$ is greater than the tss

at epoch t signifies a local minima in the tss. This may not necessarily be the absolute minimum over the attribute space but is used as the point at which to stop training the neural network.

3. If the epoch is less than 50 find the next epoch with a minimal error on the stopping set. This becomes the stopping point.

4. Once the stopping point has been established the weights of the neural network are fixed.

4.6 Choosing a Neural Network

Back-propagation networks can show substantial variation in their outputs as a result of the random selection of the initial weights. Different local minima may be found that result in poor classifiers. For this reason five networks, each with different initial weights, were trained for each classification task. This leaves us with the problem of which specific network to use for a given classification task.

An important result of this work was that networks trained as described here showed only a small variation in the range of output values. That is, these networks are less likely to find a local minima that results in a poor classifier. Thus, the specific network that is used in classification is the one that minimises the number of misclassifications by minimising the the number of false positives and false negatives.

A simple program was written to test a number of thresholds for each network and minimise the misclassified pixels. That is, we can automatically choose the best performing network for use.

4.7 Classification Accuracy

Once we have a trained classifier how can we be sure that the test set error is a reasonable estimate of the overall error of the classifier on unseen data?

A commonly used technique for assessing the true accuracy of classifiers is cross-validation [85, 173, 45, 172, 16]. Cross-validation splits the training data into different partitions, with each set of partitions being used to train a classifier.

In the case where different partitions of the data result in similar classifications and error rates, the error rate is considered a reasonable estimate of the error on unseen data.

Cross-validation used in association with neural networks requires a much larger number of networks to be trained. For each partition of the data we need to train a number of networks each with different initial weights, significantly increasing the training time required.

When all networks trained have similar error rates, either by training a number with different initial weights or using cross-validation, the networks are considered to be reasonable classifiers and the data is a reasonable representation of the concepts attempting to be captured by the classifier.

All networks, with the topologies given here and trained as described, displayed consistent results in classification accuracy for any given classification task. That is, the networks used here showed less variation in output values for specific inputs, and gave similar overall classification accuracies. Experimentation showed that training a number of networks, with different initial weights and simplifying the neural network topology as described here, achieves the same outcome as would have been achieved with cross-validation.

As cross-validation did not contribute to an improved estimation of classification accuracy it was not used in this work for neural networks. Accuracy assessment was based only on the accuracy of the five networks trained for each task. The network with the minimum number of false positives and false negatives was chosen for use.

4.8 Conclusions

Neural network classifiers are extremely flexible in how they can be used, which can mean that classification is difficult to automate. We have however limited the neural networks and given heuristics for training them that allow us to automate classifications. In summary, these are as follows.

- We restrict the topology of the networks to a three layer network that only does a binary classification. The single output node is used to determine

class membership for a single class. A different network is trained for each class to be identified.

- Output values are thresholded to minimise the number of false positives and false negatives.
- A set of heuristics were determined through experimentation that allow the stopping point in training to be determined automatically.

It must be noted that a number of network topologies were investigated before finalising this particular topology and training strategy. Most importantly no other topologies were found that gave consistent or better results. That is, networks having other topologies and configuration parameters, trained with different initial weights but using the same data, generally resulted in significantly different classifications for individual cases and often significantly different overall error rates.

While it can be argued that more optimal strategies may exist, we did not find other topologies that resulted in more accurate classifiers. Throughout this thesis we will be looking at other techniques to improve classification accuracy, rather than trying to fine tune the network topology to be optimal for the given classification task. However, the techniques given here do not exclude the possibility of modifying network configurations where appropriate or necessary.

A key result of this work is that the networks trained with this configuration give robust results without the overhead of investigating a large number of topologies and training strategies for each new data set being investigated. This is a significant step forward in automating classification.

Chapter 5

Automatic Generation of Attributes from Image Data

It is widely accepted that remotely sensed data alone is not enough to produce reliable vegetation maps. Information such as aspect, slope and climate can help determine which species will grow in a given location. However, in practice this information may not be available or if it is can be highly subjective and even erroneous. We do, however, need to ensure that the maximum amount of information is extracted from the data that is available.

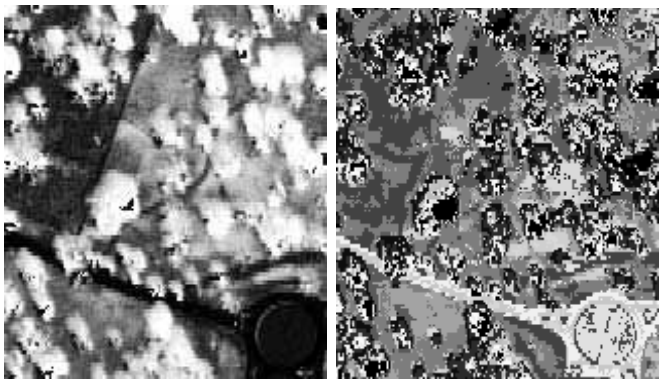
The use of data from multiple sensing platforms has been widely reported in the literature. In one example, both SAR and Landsat TM data were used to increase the differentiation of broad leaf and conifer classes [175]. The use of multi-sensor information resulted in an increase in the accuracy of the classification through better differentiation of the classes.

A variety of pre-processing techniques are available that remove noise from an image and so improve classification accuracy [134]. However, rather than trying to extract as much information from the data as possible, classification is often done directly from the image after only limited pre-processing.

Rather than using a very limited range of pre-processing and classification techniques we can instead apply a large number of such techniques to the data which can be used as additional attributes in a classification task. Generating a large number of additional attributes can be used to highlight many different features captured in the image and so will help us to improve classification accuracy.

5.1 Using Multiple Attributes to Highlight Information

Maloof et al [98] demonstrated that useful generalisation is possible when using data that differs in location and aspect. Specifically on the data used here, Figure 5.1 is an example of how different techniques can highlight different information in an image.



(a) (b)

Figure 5.1: Highlighting different features in an image using different techniques.

The classification in Figure 5.1(a) shows quite clearly a fence line (diagonal line from the top down, in the top left quadrant) that is the boundary between a paddock that is grazed and one that is not. However, the classification in Figure 5.1(b) more clearly delineates the road and roundabout in the image (along the bottom half of the image), which is not as clearly differentiated from the grazed paddock in the other classification. Note that both classifications were generated from the same spectral data.

If either technique had been used alone information that may help to distinguish between objects would not have been available. In particular, techniques that simplify the information in an image result in loss of information. However, a variety of techniques applied to an image and used as additional attributes can serve to highlight or distinguish between different features and so improve the quality of the maps produced when used in combination.

We need not limit ourselves to just pre-processing techniques. When available,

images from a variety of sensing platforms with different acquisition dates can be used. In fact, an attempt should be made to make use of any available data.

5.2 Generating Attributes

Additional attributes used for classification were generated for this work using a variety of traditional remote sensing techniques and unsupervised classification algorithms. Many more techniques could have been used, however those used for the work described in this thesis are outlined in the following sections.

5.2.1 Pre-processing Remotely Sensed Data

One of the simplest ways to highlight information in a remotely sensed image is to use simple transformations of the existing spectral data. The techniques used in this work are as follows.

Proportional Spectral Data Determining which spectral bands show the most reflectance can help to distinguish between objects. For example, vegetation will show low values in the red part of the electro-magnetic spectrum and high values in the infrared part of the spectrum [164]. We can emphasise the relative reflectance values by generating the proportional reflectance values over all bands.

The total reflectance R for a given pixel is given by

$$R = \sum_{b} r_{rc,b}$$

where $r_{rc,b}$ is the reflectance value for a pixel in row r and column c from spectral band b. A new band of data is generated from each of the original spectral bands where each pixel is assigned the value

$$\frac{r_{rc,b}}{R}$$

Averaging Spectral Data We can incorporate texture information by calculating the average reflectance values [2]. An additional band of information can be generated from each of the spectral bands where each pixel is given

the average value of all pixels in an $n \times n$ radius immediately surrounding it.

That is, the pixel in row r and column c of a texture image gets the value

$$\frac{\sum_{i=-1}^{1} \sum_{j=-1}^{1} r_{(r+i)(c+j)}}{9}$$

where r_{ij} is the reflectance of the pixel in row i and column j in the given spectral band.

A possible variation on this is to calculate the variance of the pixels values in a given neighbourhood.

5.2.2 Principal Components Analysis

Principal components analysis [134] of n spectral bands produces n linearly independant bands. This is done by transforming the n bands into a new co-ordinate system in n-space. The first component will contain most of the variation between the n bands, while the last component will contain mostly noise.

The spectral values of each pixel (i, j) can be described using a n-dimensional vector $\mathbf{x}_i, j$, where n is the number of spectral bands. If the total number of pixels is K then we define the mean vector and covariance matrix as follows.

$$\mathbf{m} = \frac{1}{K} \sum_{i,j=1}^{K} \mathbf{x}_i, j$$

$$\mathbf{C} = \frac{1}{K-1} \sum_{i,j=1}^{K} (\mathbf{x}_i, j - \mathbf{m})(\mathbf{x}_i, j - \mathbf{m})^t$$

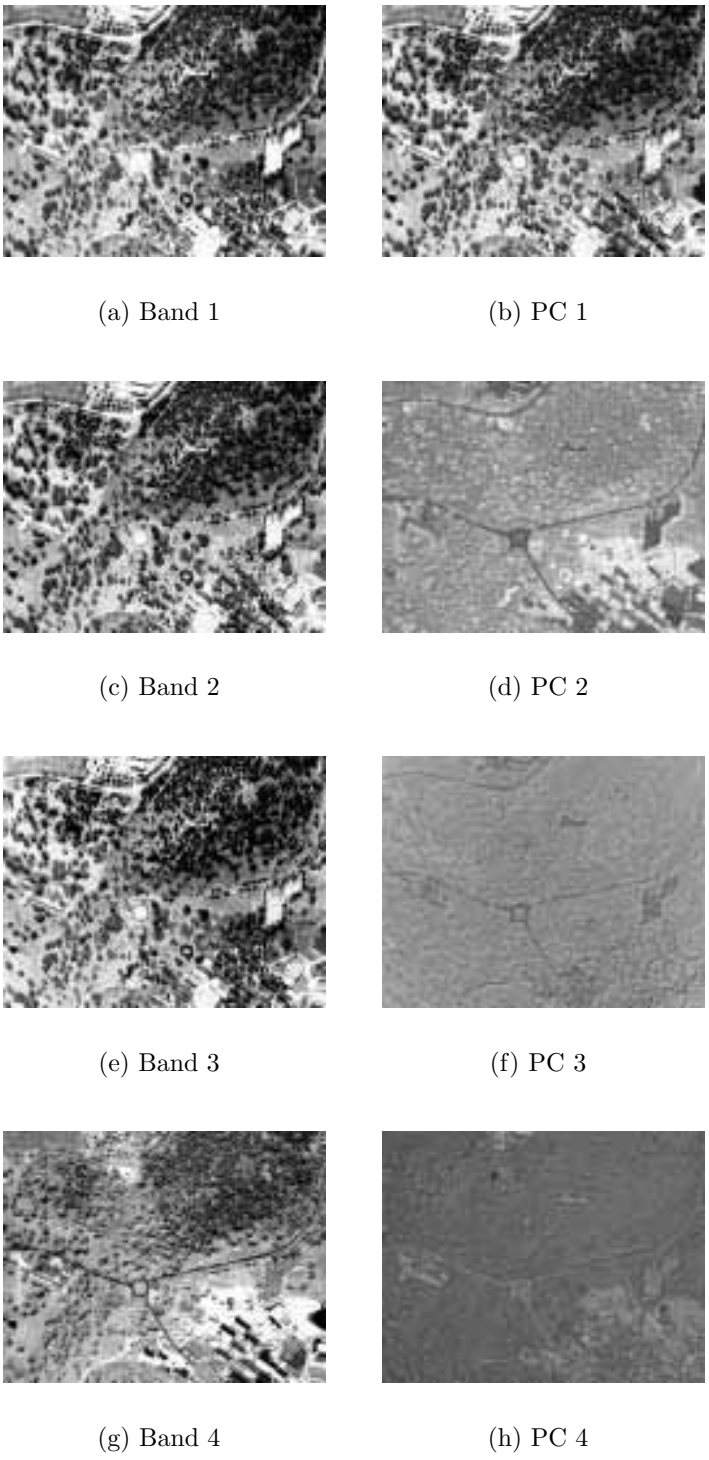
where $\mathbf{v}^t$ is the transpose of vector $\mathbf{v}$.

The transformation applied to the n bands is given by $\mathbf{y} = \mathbf{G}\mathbf{x}_m$ where each column i in $\mathbf{G}$ is the ith eigen vector.

An example of the result of applying principal components analysis to the *CSU* airborne video image is shown in Figure 5.2. Clearly the majority of the information is represented in the first component. The second component has more

clearly distinguished the urban features, such as roads and buildings, as well as the lawn surrounding them. The last two bands contain mostly noise, though this will not always be the case.

All principal components analysis for this work has been carried out using the GIS package GRASS [169].



(a) Band 1 (b) PC 1

(c) Band 2 (d) PC 2

(e) Band 3 (f) PC 3

(g) Band 4 (h) PC 4

Figure 5.2: Original spectral bands compared with the principal components analysis for the *CSU* image.

5.2.3 Vegetation Indices

Vegetation indices have been widely used in mapping applications to highlight vegetation characteristics in remotely sensed data. They can be used as additional attributes and are generated from combinations of differences, ratios, products and sums of spectral values for each pixel in an image. Differences can be useful for highlighting changes between two images over the same area. Ratios between two bands can be used to reduce the effects of topography and enhance subtle differences, such as between rock and soil [134].

The red (RED) and near–infrared (NIR) bands are most commonly used for vegetation indices. The reflectance in the red band is inversely proportional to the amount of chlorophyll since it is absorbed by the chlorophyll [164]. Solar radiation in the near–infrared is not absorbed by the vegetation, but rather is scattered and the measured reflectance values depend on the reflectance of the materials in the image [164]. That is, both the red and near–infrared bands are sensitive to vegetation [164], and so the name vegetation indices.

The simplest and most commonly used vegetation indices are, the *ratio vegetation index* (RVI) given by

$$\frac{NIR}{RED}$$

the *difference vegetation index* (DVI) given by

$$NIR - RED$$

and the *normalised difference vegetation index* ($NDVI$) given by

$$\frac{NIR-RED}{NIR+RED}$$

These indices aim to take advantage of the strong contrast between the reflectance of green healthy vegetation in the visible and near–infrared band and the lack of contrast in these bands for soils [127].

The $NDVI$ ranges in value from -1 to 1, with values around -1 indicating clouds, water or snow, 0 rock or soil and 1 vegetation. The closer the $NDVI$ is to 1 the more vegetation there is.

The $NDVI$ has been useful in monitoring vegetation as it is relatively insensitive to changes in, for example, illumination, slope and aspect [95]. The RVI helps reduce the effects of changes in illumination over the image due to the sensor or topography. However, quantitative interpretation of both the $NDVI$ and RVI is difficult as they are sensitive to the conditions in the atmosphere and the geometry of illumination and observation [128]. The RVI, $\sqrt{RVI}$ and DVI have also been found to be useful for monitoring plant development [164].

The *transformed normalised difference* (TVI) [72] given by

$$\frac{\sqrt{NIR-RED}}{(NIR+RED+0.5)}$$

has been used to monitor rangelands and wheat crops as it shows a high correlation with green biomass and so is useful for monitoring plant growth [164].

To monitor changes in the growth of a plant the $GREEN/RED$ ratio has been found to be useful although the RED and NIR combinations appear to be superior [164].

Variations in the rock or soil brightness will have a large effect on the above mentioned indices [47]. Dark backgrounds will cause an overestimation of vegetation when compared with bright backgrounds [47]. The background soil effects can influence the values produced by a vegetation index and so reduce its effectiveness in predicting vegetation cover.

Indices such as the *perpendicular vegetation index* (PVI), *green vegetation index* (GVI), *soil line index* (SLI) and *soil adjusted vegetation index* ($SAVI$) have been used to detect changes in green vegetation while holding the soil background constant [72, 131]. The disadvantage with the PVI, GVI and SLI, in the context of this work, is that they all depend on ground truth data. That is information about the soil characteristics are taken into account.

The $SAVI$, defined as

$$\frac{0.5(NIR-RED)}{(NIR+RED+0.5)}$$

lead to the *modified soil adjusted index* ($MSAVI$) being proposed [131] to increase the sensitivity to vegetation as well as further reducing the soil effects.

$$\frac{2.NIR+1-\sqrt{(2.NIR+1)^2-8(NIR-RED)}}{2}$$

A major problem with vegetation indices is that they are sensitive to the properties of the background materials and can produce incorrect results [46]. Greenness measures are highly dependant on the soil brightness [72].

Crippen [38] suggests that the popular vegetation indices are used, at least in part, because of the success of past work rather than looking at possible alternatives. There are however a potentially infinite variety of vegetation indices.

For the Australian semi–arid regions the commonly used vegetation indices have been shown to be inappropriate [122]. O'Neill [122] found that there is a need to explore the potential of vegetation indices using high spectral and spatial resolution data under Australian conditions as little has been done. She also states that different indices are appropriate in different seasons.

The $NDVI$ and RVI have been found to have limitations in semi–arid regions due to the effects of soil background. O'Neill found that the *stress related index* (SRI)

$$\frac{MIR.RED}{NIR}$$

using the mid-infrared, near-infrared and red spectral bands, was useful in monitoring semi–arid vegetation which was strongly related to the vegetation cover in both winter and summer.

5.2.4 Using Multiple Vegetation Indices

Malthus et al [99] tested combinations of ratios and the $NDVI$ for a number of mid-infrared (MIR) bands. A number of vegetation indices were found to be least sensitive to soil effects and gave reasonable estimates of canopy cover.

This seems to imply that a combination of vegetation indices may be useful in generating evidence for the occurrence of a particular class. Each of the vegetation indices used can serve to compensate for the inadequacies of the others, and highlight different information within the data.

Little work appears to have been done in comparing many combinations of bands for a large number of different indices. With the more recent advances in sensor technology very detailed spectral data is now available. For example, the AVIRIS system provides 224 spectral measurements per pixel [130]. This means a far richer set of vegetation indices is possible, but also that more computationally expensive investigations of combinations will need to be carried out.

There is no reason to limit ourselves to the use of a single vegetation index. If we generate a number of different vegetation indices using all possible combinations of bands we are extending our ability to find subtle information in the data. We are also better able to find information that can help distinguish between two different objects that may otherwise have been confused.

A common theme when using specific indices for specific classification problems is that they are sensitive to noise of various kinds. That is, when the environment contains certain elements it can bias the results of the index. However, if a vegetation index is not used in isolation, as an end in itself, we can reduce the effects of the noise.

For the purposes of this work the vegetation indices used have been limited to those described in this chapter, though many more could be used. A large number of indices were generated using all possible combinations of spectral bands for each type of vegetation index listed in this section. For example, with an ABVS image with four bands we would generate all six combinations of bands using the ratio vegetation index - $\frac{red}{blue}$, $\frac{red}{NIR}$, $\frac{red}{green}$, $\frac{blue}{NIR}$, $\frac{blue}{green}$, $\frac{NIR}{green}$.

5.2.5 Unsupervised Classification

As discussed in Section 2.4.10, unsupervised classification aims to group pixels with similar properties. This can be useful for finding novel or interesting features in the data that would not otherwise be found. It can also be used to highlight significant information by reducing the dimensionality of the data.

Unsupervised classifiers used in this work to generate additional attributes were AutoClass and unsupervised maximum likelihood. The main advantage of AutoClass is that it settles on a number of classes based on the characteristics of the data. Unsupervised maximum likelihood classification was carried out using the GIS package GRASS [169] and required the number of classes to be specified by

the user. For this reason a number of maximum likelihood classifications were generated.

5.2.6 Incorporating Contextual Information

To date much of the classification of remotely sensed data has been with attribute-based techniques. However, it is well known in the field of machine learning that attribute value learning is not always sufficient. This sentiment is also expressed within the remote sensing community. In remote sensing and vegetation mapping domains it would be unusual for a pixel to have a class independantly of its neighbours. The difficulty is how to incorporate this contextual[1] information into a classification.

A number of methods are available for incorporating contextual information. For example, relaxation labelling [134] takes a classified image and updates the probabilities of class membership for each pixel using the class probabilities of neighbouring pixels. Ahuja [2] found that using the surrounding $n \times n$ pixel values as additional attributes for training a classifier was more useful than using attributes derived from the contextual information.

Simple ways of including context in an unsupervised, or even a supervised, classification are as follows.

- Include the values for a given attribute in an $n \times n$ region surrounding a pixel in the list of attributes for that pixel.
- Take the classes[2] from a classified image in an $n \times n$ region surrounding a pixel as the attributes for that pixel.

[1] In remote sensing and vegetation mapping domains this is referred to as spatial information.

[2] That is, once a classification of an image has been generated using a supervised or unsupervised classification technique each pixel in the image will be given a class label. In the case of a supervised classification this label already maps directly to an identifiable real world concept like "grass" or "building". In the case of an unsupervised classification the class given would be based on a measure of similarity between pixels and each label can be mapped to a real world concept using expert knowledge. Typically a classification will serve to reduce the dimensionality of the data and simplify it into a smaller number of classes. Where a classification is being used in further classification tasks (i.e. it becomes an attribute in another classification) a smaller number of classes means a smaller number of possible values for the given attribute. Labels generated by such classification techniques can be considered an additional attribute of a given pixel so used in further classification tasks.

- Take the counts of pixels in each class from a classified image in an $n \times n$ region surrounding a pixel as the attributes for that pixel.
- Weight the attribute values to be considered in an $n \times n$ region surrounding a pixel. That is, the further away the context pixel is the less influence its class has.

Incorporating context in this way can generate a large number of additional attributes very quickly as the number of attributes, classes or radius of context pixels n increases. However, to keep the number of attributes used in a given classification within reasonable bounds unsupervised classification was carried out on an $n \times n$ region for a single attribute using AutoClass. This classification then became an additional attribute for a supervised classification.

5.3 Discussion and Conclusions

Using the techniques described in this Chapter we can clearly generate a large number of additional attributes. Such attributes are commonly used in remote sensing applications because they serve to highlight particular features in a remotely sensed image and help to remove irrelevant information.

A large number of attributes can be generated just from contextual information. The spectral values from each of the eight neighbouring pixels could be used as additional attributes for a given pixel or used in the generation of further attributes that can be presented to a classifier. While such approaches are possible the increased data dimensionality does increase the amount of information being presented to a classifier and consequently increases the training times dramatically. As there are already a large number of attributes being investigated neighbouring pixels were not were not considered here.

The techniques used to generate additional attributes need not be limited to those discussed here. It is possible to generate a vast number of attributes from a single image, even if we limit ourselves to the techniques described here. The transformations described here are only a small selection of those that can be used and were chosen as they are commonly used in remote sensing domains.

The problem now is that using all of these techniques will certainly result in the generation of a large number of possibly irrelevant or overlapping attributes. For

a specific classification task not all attributes that can be generated will be useful and will only serve to obscure the most relevant information in the available data. This additional data can swamp a given classifier and end up reducing classification accuracy. In fact, we use the classifiers themselves to learn which attributes contain information of relevance and which do not. This will be addressed, using attribute selection, in the following two Chapters.

It must also be reiterated that the techniques being discussed in this thesis, including generation of a large number of attributes, is not being recommended as a general technique for all datasets. The domains this work should be applied to are those where only small, noisy datasets are available. If a large dataset is available, that has an information dense set of attributes to work with, then this approach would most certainly be overkill.

As we will see identification of the most relevant attributes of those generated means that the classifier can be trained on a significantly reduced set of attributes. Training times at this point become considerably faster, as do classification times, and the overall error rates can be reduced.

Chapter 6

Attribute Selection

Attributes used in classifications can contains errors, due to inaccuracies in collecting data or noisy information. This can mean that attributes, that in theory, should be useful can be rendered useless. An example of this is climate data, which is often generalised and so does not take into account local variations or microclimates. This makes such data of questionable use for detailed species level vegetation maps.

Additional attributes can be generated using a variety of techniques, as discussed in the previous chapter. By generating additional attributes from the existing data we can highlight significant features in an image and so improve classification accuracy. However, training a classifier with a small number of cases and a large number of attributes we may over-fit[1] the training data [34, 56]. In addition a large number of attributes takes longer to train the classifier and may generalise poorly [129]. That is, with a large number of attributes we either run the risk of learning the details of the training set, or not learning enough of the general properties of the training set, and so are not able to predict anything about unseen data with any reasonable degree of accuracy.

From the large number of possible generated attributes, along with those originally available we need to find the most useful for a given classification. We also need to be able to do this automatically.

[1]To over-fit a classifiers is where the classifier can not return meaningful classifications for unseen data unless it is almost identical to the original training data. This makes the classifier all but useless in classification tasks.

6.1 Previous Work

Identifying irrelevant attributes, to remove them from a given classification task, is a major area of concern in machine learning [89]. To enable useful predictions to be made a learning algorithm needs to be presented with the attributes that are most relevant for a particular domain.

Langley and Sage [90] showed that naive Bayesian classifiers are sensitive to redundant attributes. Chen et al [24] were able to achieve higher classification rates using C4.5 after attribute selection was carried out. Work carried out as part of this thesis (see [111], and discussed later in this and the following chapter) demonstrated improved performance in neural networks after attribute selection was carried out.

Grigg et al. [67] used the wrapper technique with C4.5 to select the most relevant attributes, derived from a spectrogram. From a set of 70 attributes as identified by a group of experts, only 15 were chosen for use by C4.5. Reducing the number of attributes in this way contributes to faster processing times in a trained classifier.

Induction algorithms, such as C4.5, generalise poorly if allowed to use all available attributes as compared with using a good subset of attributes [20]. Poor generalisation is only exacerbated when the attribute values are also noisy. In particular, attribute selection in a vegetation mapping domain becomes important when we consider that hyper-spectral data is now becoming available (up to 255 bands) [76] and other derived or surveyed data may be noisy.

By selecting the most relevant attributes we can potentially show a more direct relationship between the attributes and the objects on the ground [39]. Crist et al. [39] used tasseled cap transformations to capture 95% of the variability in an image using half the number of bands for Landsat MSS and Landsat TM data. The tasseled cap transformations adjust the viewing perspective of the data such that temporal sequence information is highlighted.

One of the main problems with supervised classification in a vegetation mapping domain is the small amount of training data. Classification accuracy will start to decrease as the ratio of training cases to attributes decreases [76]. So, if we have a large number of irrelevant attributes our ability to produce a reliable classification is reduced. On the other hand, the more classes a classifier needs to identify the more attributes that are needed [76]. Thus, there is a trade off between the

number of attributes and the number of training cases and classes, requiring us to identify the most relevant attributes for a given classification task.

6.2 Attribute Selection Algorithms

In general, an exhaustive search for the best set of attributes is not possible as there are $2^n - 1$ subsets of n attributes. So a number of heuristic search techniques have been developed.

The search for a set of attributes can use a number of approaches, starting with an empty set of attributes and successively adding attributes, or starting with all attributes and successively removing them [89]. Often greedy search is used – one attribute is considered at a time, rather than groups of attributes – to produce a subset of good attributes. In the case of greedy search, once an attribute has been chosen it is never reconsidered.

Evaluation of the search is often measured by the error on a test set for a given classification technique. The search can be stopped when the error rate on the test set stops improving, continuing while the error rate does not degrade or investigating all possible subsets and choosing the one with the best error rate [89].

The FOCUS algorithm [5] exhaustively examines all subsets of the attributes for the minimal subset that is sufficient to determine the class of a given input case. The Relief algorithm [82] uses a measure of relevance for each attribute to do attribute selection.

John et al. [77] argue, however, that the attributes selected should depend not only on the attributes and classification task but also on the classification system used. To this end, they proposed the wrapper method which uses the classifier itself to do the attribute selection. Starting with an empty set, attributes are added and the accuracy of a classification from these attributes is tested. This process continues until the accuracy does not improve. Cross-validation can be used to test classification accuracy. This search method can be further improved by considering deletion of attributes at each step as well.

Attribute selection for different classifiers has shown that different subsets of attributes can be more relevant to a given classifier, for the same classification task [77, 76]. The key point to note about the wrapper method is that selecting

appropriate attributes is done by the classifier to be used.

Not all datasets will have irrelevant attributes as an expert working in a given domain will tend to chose only those attributes for inclusion in a dataset that are relevant. In addition to this, the majority of datasets contain a large number of training cases. For example, the UCI machine learning database, which contains a large number of datasets for testing machine learning techniques, tends to contain only relevant attributes [89]. In this case attribute selection may not be relevant.

In this thesis we are investigating datasets with potentially irrelevant attributes and we are generating a large number of additional attributes. The number of attributes used, and so the number of irrelevant attributes, is then decreased by carrying out attribute selection. This approach is only feasible if the task can be automated, which can be achieved by using the wrapper attribute selection technique.

The wrapper technique is a significant improvement over simply verifying the error rates for all possible subsets of the available attributes. The computational requirements for this algorithm are $O(\frac{n(n+1)}{2})$, where n is the number of attributes. The worst case is where one classifier is required to be trained for each possible subset of the attributes. In practice the number is substantially less than this, particularly when there are noisy or irrelevant attributes.

When using the wrapper with neural networks the additional training overhead is generally too high. The overhead is created by having to train additional networks for each subset of attributes with different starting weights, to ensure that a local minima is not found. This increases the worst case number of classifiers to $m\frac{n(n+1)}{2}$, where m is the number of networks trained for each subset of attributes. As training times are already significantly higher for neural networks than for many other classifiers an alternative would be preferable. A solution to this problem is introduced in Chapter 7.

6.3 Attribute Selection in Remote Sensing Domains

Typically only a relatively small number of attributes is used for a given problem in mapping from remotely sensed data. Attributes seem more often to be generated

and then selected based on expert knowledge, rather than from extensive searches, as is often the case in machine learning domains. In the past this has been due, in part, to limited computational resources. It is now feasible to explore considerably larger data sets automatically.

Much work has been done on comparing vegetation indices, as outlined in Section 5.2.3. For these problems vegetation indices are chosen based on their ability to distinguish specific characteristics of the area being studied.

One approach to attribute selection used in the remote sensing domain has been to remove redundancy using separability measurements, such as the Bhattacharryya distance, so that the selected attributes provide class separability [78].

A commonly used approach to reduce the number of attributes is by using principal components analysis, a technique that transforms the data into a different attribute space (see Section 5.2.2). The last few components are removed as most of the variation in the data is contained in the earlier components, however, it is possible for subtle details to show up in the later components. Such techniques serve to remove noise from the data as well as reduce the number of bands to be considered [134, 60, 97], but once again are difficult to apply to data sets that include non-numeric data.

Conese and Maselli [32] investigated the use of mutual information analysis to select the optimum bands for a given classification task. Mutual information analysis is a statistical technique that evaluates the probabilistic information common to different variables. This can be used to determine which bands contain the most information for a given classification task. Techniques such as this are particularly useful when trying to reduce the number of irrelevant bands without transforming the original spectral data, as would happen with principal components analysis. Using the optimum subset of spectral bands from such analysis increased classification accuracy by around 10%. However, there were still problems with the effects of topography and illumination, for example.

Attribute selection in a remote sensing domain has also been carried out to not only achieve more efficient classifications but also to display images to their best advantage. For example, a false colour composite maps the green, infrared and red bands in an image to the colours blue, red and green respectively. However, such displays are used largely to highlight information for human rather than computer interpretation.

Overall, attribute selection appears not to have been investigated in this domain to the same extent as for machine learning as it has relied more heavily on expert evaluation. Now with the widespread availability of remotely sensed data and computing systems, systematic and extensive investigations can be carried out automatically.

However, it must be kept in mind that background knowledge is also critical in making sense of the attributes being chosen from the attribute selection process. Training data may not be truly representative of the classes and so not be able to fully make use of the relationships between the attributes used in classification.

6.4 Conclusions

One aim of this thesis is to select the attributes that are most useful for a given classification task from a large set of attributes that have come from either existing data for an area or have been generated from existing data. Generating a large number of attributes allows us to highlight a large number of features in the data and attribute selection allows us to choose the most relevant attributes for a given classification task.

The wrapper is a relatively simple technique for attribute selection and can be used with any given classifier. The main advantage of this technique is that the attributes are chosen by using the classifier itself. As different classifiers use different properties of the data, different subsets of the attributes may be more appropriate for different classifiers.

The wrapper method of attribute selection is used for C4.5 and IBL classification tasks discussed in this thesis. However, as the wrapper is such a computationally expensive technique when used with neural networks an alternative is demonstrated in the next chapter.

Chapter 7

Attribute Selection for Neural Networks

The wrapper method of attribute selection is not ideal when applied to neural networks as the computation time becomes intractable for any reasonably sized dataset. As attribute selection is central to this work an alternative for neural networks needed to be found.

The technique developed, called contribution analysis, used the weights of a trained neural network to determine the contribution of each input to the output of the neural network. Initial experiments showed this was a useful measure for determining which of the attributes are the most relevant in a given classification task. An advantage of this technique over the wrapper is that it only requires one additional training run of the neural network and provides similar results to the wrapper, which is demonstrated in this chapter. The use of this technique has also been discussed in [110].

As contribution analysis is a new technique we diverge temporarily from the main theme of this thesis to demonstrate its use as a general purpose attribute selection technique. The results of its application to a number of different datasets in different domains was investigated. Neural network classification after contribution analysis attribute selection was compared with classification after wrapper attribute selection. Both attribute selection techniques were also compared to a straight classification using all attributes for each dataset, using both neural network and C4.5 classification.

7.1 Assigning Contribution to Attributes

The networks for which the contributions of inputs are determined, consist of three layers, an input layer (with *ninputs* inputs), a single hidden layer (with *nhidden* nodes) and an output layer (with *noutputs* nodes). The input nodes are numbered from 1 to *ninputs*, the hidden nodes are numbered from $ninputs + 1$ to $ninputs+nhidden$ and the output nodes are numbered from $ninputs+nhidden+1$ to $ninputs + nhidden + noutputs$. The connection weight between node i in layer n and node j in layer $n+1$ is given by w_{ji}.

Garson [62] proposed the following measure of contribution. The contribution that input node i makes to output node o is

$$\frac{\sum_{j=1}^{nhidden} \frac{w_{ji}}{\sum_{l=1}^{ninputs} w_{jl}}.w_{oj}}{\sum_{k=1}^{ninputs} \left(\sum_{j=1}^{nhidden} \frac{w_{jk}}{\sum_{l=1}^{ninputs} w_{jl}}.w_{oj} \right)}$$

This method will not give a true proportion when there is a combination of positive and negative weights. Another measure of contribution, used in [177], gives the contribution of a node in one layer to a node in the next layer. For example, the contribution of input node i to the hidden node h would be

$$\frac{|w_{hi}|}{\sum_{l=1}^{ninputs} |w_{hl}|}$$

A disadvantage of this in its current form is that the sign of the contribution is lost, and so a true proportion is not given.

The measure used for this work (also discussed in [110]) is a modification of the original measure given by Garson and instead, defines the contribution of input node i to output node o as follows.

$$\frac{\sum_{j=1}^{nhidden} \frac{w_{ji}}{\sum_{l=1}^{ninputs} |w_{jl}|} . w_{oj}}{\sum_{k=1}^{ninputs} (\sum_{j=1}^{nhidden} |\frac{w_{jk}}{\sum_{l=1}^{ninputs} |w_{jl}|} . w_{oj}|)}$$

Figure 7.1: Contributions for each input to the output of the neural network.

When defined in this way weights can be positive or negative and we will still obtain a true proportion.

7.2 Attribute Relevance

John, Kohavi and Pfleger [77] define attributes as being

Strongly relevant if the attribute is necessary and can not be removed without decreasing the number of correct classifications

Weakly relevant if the attribute sometimes contributes to the classification

Irrelevant if the attribute will never contribute to the classification

How to apply these definitions to contributions of attributes was investigated as part of this work. The initial investigation of identifying the relevance of attributes was also published in [110].

To use the contribution for attribute selection, the contributions of all inputs of a single neural network can be calculated and those with *small* contributions can be discarded as irrelevant.

In addition, the consistency of the contribution was investigated. As neural networks are trained with random initial weights a minimum error network may be found that does not provide a meaningful or useful classification. When a number of networks are trained the contributions of individual attributes will show at least some variation. The hypothesis was that attributes with *large* variation in contributions across a number of networks could also be removed as there is no consistent information content in the available data.

Therefore, the approach investigated to measure the relevance of attributes from the contribution analysis was as follows:

- Large contributions with small variations across all trained networks determine the strongly relevant attributes.
- Small contributions with small variations across all trained networks determine the irrelevant attributes.
- Large variations with contributions clustering around zero determine weakly relevant or irrelevant attributes.
- Large variations with large contributions determine weakly relevant attributes, but in some cases may be strongly relevant attributes.

The experiments carried out in this work, and discussed in this chapter demonstrate what constitutes a *large* or *small* contribution and a *large* or *small* variation.

7.3 Using Contributions for Attribute Selection

The relevance of attributes, giving their contributions and the variation in that contribution, can be visually represented by plotting the contribution of each attribute for each of the networks trained (see Figure 7.2). Each point in the plot shows the contribution of a single attribute for one of the neural networks trained on the given classification task.

Figure 7.2 shows an idealised situation for a three input network. Each attribute of the classification task being mapped to an input node of the network. For each input node in each network trained the contribution to the output node is

calculated using the formula in Figure 7.1, and then all contributions are plotted on the graph.

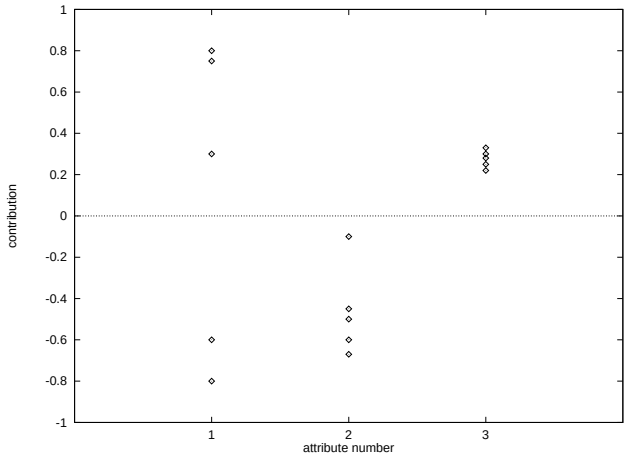


Figure 7.2: Plotting contribution of each attribute.

The graph in Figure 7.2 shows that attribute 1 has a large variation in the contribution, while attribute 3 has a small variation. Attribute 2 shows a large contribution with variation. A small contribution is one where the values all cluster closely to zero.

The results reported in [110] removed attributes that showed:

- a small contribution, that is, attributes whose contributions clustered around zero, and showed little variation
- a large variation in contributions across each of the networks trained

When these attributes were removed, an improvement in classification accuracy was achieved. In particular, if too many attributes were removed the classification accuracy decreased, as would be expected.

We extend the work discussed in [110] further by investigating how contribution analysis compares with the wrapper for attribute selection, as well as determining what constitutes an irrelevant attribute.

7.4 Evaluation of Contribution Analysis

The following classifications were carried out and the results compared.

- neural network using all attributes
- C4.5 using all attributes
- neural network with wrapper attribute selection
- C4.5 with wrapper attribute selection
- neural network with contribution analysis attribute selection

The default settings were used for the C4.5 classifications and the neural network configurations used were as described in Chapter 4.

The datasets used were selected from the UCI Machine Learning Repository [170]. The training data for each dataset was split into three sets of roughly equal size using random selection. The training set was used to train the classifier. The stopping set was used only for the neural network to determine when to stop training as described in Chapter 4. The test set was not used at any time during training and was used to give an unbiased estimate of the error for the classifier.

Each classification task was formulated as a binary classification problem – typically this meant that a case is in a given class or not in that class. That is, where there were more than two classes for the given task all cases not in the given class were grouped into a single "not in" class.

Attributes were mapped to values between 0 and 1, and outputs to 0.1 (not in the given class) or 0.9 (in the given class). An input case was classified as not in the given if the output of the neural network was less than 0.5 and in the given class otherwise.

7.4.1 Iris Flower Data

The iris dataset (*IRIS*) has been widely used for evaluating classification algorithms. Training cases were given for three types of iris – *SETOSA*, *VIRGINICA* and *VERSICOLOR*. The attributes used to describe the flowers were

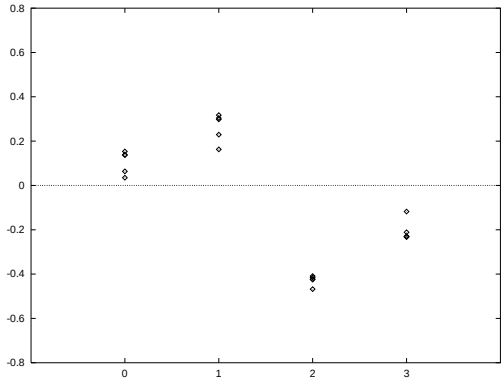
Attribute Number	Attribute Name
0	`sepal-length`
1	`sepal-width`
2	`petal-length`
3	`petal-width`

where each attribute was measured in centimetres.

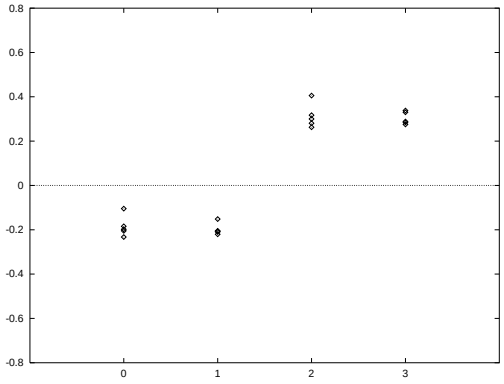
Fifteen networks, five for each type of iris, were trained. All training cases were used for each of the networks with the class of each case mapped to either 0.1 or 0.9. Target classes were mapped to 0.9 for an iris of a given single type and 0.1 otherwise. The number of cases in each of the training sets was as follows.

	Target Class	# Training Cases	# Stopping Cases	# Test Cases
setosa	0.9	17	17	16
not setosa	0.1	33	33	34
virginica	0.9	16	17	17
not virginica	0.1	34	33	33
versicolor	0.9	17	16	17
not versicolor	0.1	33	34	33

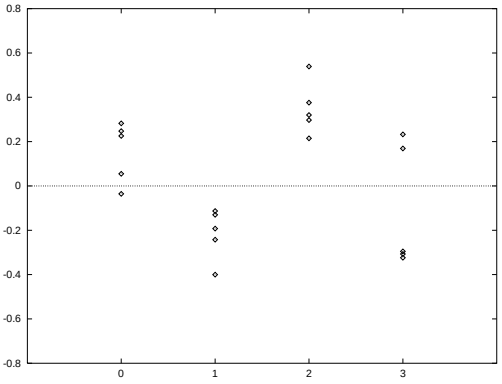
The contribution of each attribute to the output of five different networks trained for each class can be seen in Figure 7.3. For both the *SETOSA* and *VIRGINICA* class the contributions of all attributes is fairly consistent, that is, the contributions from different networks for each attribute cluster at a similar level.



(a) *SETOSA*.



(b) *VIRGINICA*.



(c) *VERSICOLOR*.

Figure 7.3: Contribution of each attribute for the *IRIS* data. Plots show attribute number vs contribution for each of the five networks.

Using the contributions of each attribute the relevant attributes were chosen. The attributes with contributions larger than 0.2 or less than -0.2 and a small variation were kept.

The wrapper attribute selection was also done for each of the datasets. The attributes selected from each of the three methods are listed in Table 7.1.

	Contribution	NN Wrapper	C4.5 Wrapper
SETOSA	`petal-length` `petal-width`	`petal-length`	`petal-width`
VIRGINICA	`petal-length` `petal-width`	`petal-width`	`petal-width`
VERSICOLOR	`sepal-width` `petal-length`	`petal-width`	`petal-width`

Table 7.1: Attributes selected for the iris data.

Next, the five types of classifiers were trained - both the neural network and C4.5 using all of the attributes, the neural network after contribution analysis attribute selection and the neural network and C4.5 after wrapper attribute selection. Each case in the test set was then classified using each classifier, a summary of the results on the test set can be seen in Tables 7.2, 7.3 and 7.4. Each table shows the number of cases for each class and the classification they were given by the classifier.

		Classified As	
		setosa	not setosa
Class	setosa	16	
	not setosa		34

(a) Neural network with all attributes.

		Classified As	
		setosa	not setosa
Class	setosa	15	1
	not setosa		34

(b) C4.5 with all attributes.

		Classified As	
		setosa	not setosa
Class	setosa	16	
	not setosa		34

(c) Neural network wrapper.

		Classified As	
		setosa	not setosa
Class	setosa	15	1
	not setosa		34

(d) C4.5 wrapper.

		Classified As	
		setosa	not setosa
Class	setosa	16	
	not setosa		34

(e) Contribution analysis.

Table 7.2: Summary of the *SETOSA* error rates for the test set, numbers representing the actual number of cases in the given class and the classification they were given.

		Classified As	
		virginica	not virginica
Class	virginica	17	
	not virginica	3	30

(a) Neural network with all attributes.

		Classified As	
		virginica	not virginica
Class	virginica	15	2
	not virginica	3	30

(b) C4.5 with all attributes.

		Classified As	
		virginica	not virginica
Class	virginica	15	2
	not virginica	3	30

(c) Neural network wrapper.

		Classified As	
		virginica	not virginica
Class	virginica	15	2
	not virginica	3	30

(d) C4.5 wrapper.

		Classified As	
		virginica	not virginica
Class	virginica	17	
	not virginica	4	29

(e) Contribution analysis.

Table 7.3: Summary of the *VIRGINICA* error rates for the test set.

		Classified As	
		versicolor	not versicolor
Class	versicolor	14	3
	not versicolor		33

(a) Neural network with all attributes.

		Classified As	
		versicolor	not versicolor
Class	versicolor	14	3
	not versicolor	3	30

(b) C4.5 with all attributes.

		Classified As	
		versicolor	not versicolor
Class	versicolor	14	3
	not versicolor	2	31

(c) Neural network wrapper.

		Classified As	
		versicolor	not versicolor
Class	versicolor	14	3
	not versicolor	3	30

(d) C4.5 wrapper.

		Classified As	
		versicolor	not versicolor
Class	versicolor	11	6
	not versicolor		33

(e) Contribution analysis.

Table 7.4: Summary of the *VERSICOLOR* error rates for the test set.

Clearly the contribution analysis attribute selection compares favourably with the wrapper technique.

For the neural network wrapper both `petal-length` and `petal-width` had a 0% error rate when used alone to train a neural network for the *SETOSA* dataset - this means that either attribute could have been chosen. The implementation of the wrapper technique used throughout this work classifies each subset and chooses the first attribute in the list that reduces the error rate. This attribute is then added to the set of good attributes. Thus, the attributes chosen will depend to some extent on their order in the search list.

The *VERSICOLOR* data set had a much larger error rate than either of the other two iris types, as shown in Tables 7.4(a) and 7.4(b). This appears to be reflected in the contributions of the attributes with a larger variation for each attribute over the five networks as seen in Figure 7.3(c).

Little or no improvement was shown in the accuracy after attribute selection for each of the classification tasks, though a smaller number of attributes reduces the classification times. That is, attribute selection has removed irrelevant or weakly relevant information without impacting the classification accuracy.

7.4.2 Mushroom Data

The mushroom (*MUSHROOM*) dataset was drawn from *The Audubon Society Field Guide to North American Mushrooms*, G.H. Lincoff, 1981. It has 22 attributes, as follows

Attribute Number	Attribute Name	Attribute Values
1	`cap-shape`	bell, conical, convex, flat, knobbed, sunken
2	`cap-surface`	fibrous, grooves, scaly, smooth
3	`cap-color`	brown, buff, cinnamon, gray, green, pink, purple, red, white, yellow
4	`bruises?`	yes, no
5	`odor`	almond, anise, creosote, fishy, foul, musty, none, pungent, spicy
6	`gill-attachment`	attached, descending, free, notched
7	`gill-spacing`	close, crowded, distant
8	`gill-size`	broad, narrow
9	`gill-color`	black, brown, buff, chocolate, gray, green, orange, pink, purple, red, white, yellow
10	`stalk-shape`	enlarging, tapering
11	`stalk-root`	bulbous, club, cup, equal, rhizomorphs, rooted
12	`stalk-surface-above-ring`	fibrous, scaly, silky, smooth
13	`stalk-surface-below-ring`	fibrous, scaly, silky, smooth
14	`stalk-color-above-ring`	brown, buff, cinnamon, gray, orange, pink, red, white, yellow
15	`stalk-color-below-ring`	brown, buff, cinnamon, gray, orange, pink, red, white, yellow
16	`veil-type`	partial, universal
17	`veil-color`	brown, orange, white, yellow
18	`ring-number`	none, one, two
19	`ring-type`	cobwebby, evanescent, flaring, large, none, pendant, sheathing, zone
20	`spore-print-color`	black, brown, buff, chocolate, green, orange, purple, white, yellow
21	`population`	abundant, clustered, numerous, scattered, several, solitary
22	`habitat`	grasses, leaves, meadows, paths, urban, waste, woods

Each attribute value was mapped to a number between 0 and 1. Five networks were trained to recognise a mushroom as poisonous or edible, with target neural

network outputs of 0.1 and 0.9 respectively. The number of cases in each of the training datasets was as follows.

	# Training Cases	# Stopping Cases	# Test Cases
edible	436	143	135
poisonous	241	83	91

We must keep in mind that the attributes used in the datasets discussed in this chapter were originally chosen because they do indeed have relevant information. In particular, the attributes used to distinguish edible and poisonous mushrooms are most likely all relevant. However, can we at least find weakly relevant attributes that can be removed?

The contribution for the five networks trained on the *MUSHROOM* data can be seen in Figure 7.4.

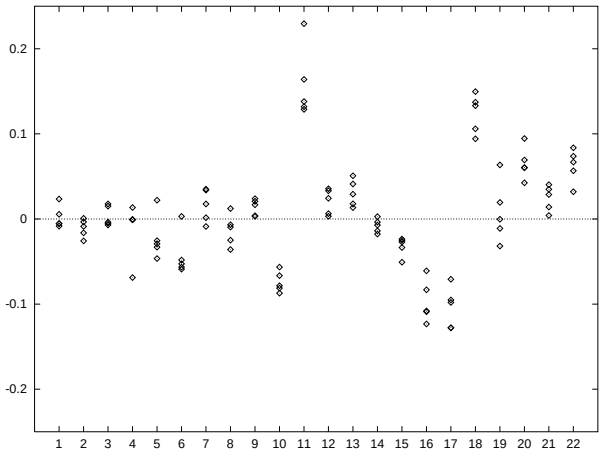


Figure 7.4: Contribution of each attribute for the *MUSHROOM* data.

Attributes that can be chosen for a large contribution are

`stalk-shape`(10)
`stalk-root`(11)
`ring-number`(18)

Those that could be selected for low variation and non-zero contribution, though not as clear, are

stalk-shape(10)
stalk-surface-above-ring(12)
stalk-surface-below-ring(13)
stalk-color-below-ring(15)
veil-type(16)
veil-color(17)

The attributes that were retained were those that did not cross the zero axis or clustered close to zero. The attributes selected by the three attribute selection methods are listed in Table 7.5.

Contribution	NN Wrapper	C4.5 Wrapper
stalk-shape	cap-shape	odor
stalk-root	cap-surface	spore-print-color
ring-number	cap-color	population
stalk-surface-above-ring	gill-size	
stalk-surface-below-ring	stalk-shape	
stalk-color-below-ring	stalk-root	
veil-type	stalk-surface-below-ring	
veil-color	veil-type	
	spore-print-color	
	population	
	habitat	

Table 7.5: Attributes selected for the *MUSHROOM* data.

The results for all of the five final classifications give a 0% error rate on the test set, except for the contribution attribute selection which had an error rate of 4%, as seen in Table 7.6(e)). It is unclear why the C4.5 wrapper should produce a 0% error, first on the training set and then as well for the test set with so few attributes. Further investigation of the data, with input from a domain expert, as well as investigation of the classification algorithm might be able to resolve this issue.

Attribute selection for the *MUSHROOM* dataset has allowed us to reduce the number of attributes with only a small reduction in the accuracy for the contribution attribute selection. In this particular case it would, however, be desirable to investigate additional subsets of the attributes to ensure that at least poisonous mushroom are never identified as edible.

		Classified As	
		edible	poisonous
Class	edible	135	
	poisonous		91

(a) Neural network with all attributes.

		Classified As	
		edible	poisonous
Class	edible	135	
	poisonous		91

(b) C4.5 with all attributes.

		Classified As	
		edible	poisonous
Class	edible	135	
	poisonous		91

(c) Neural network wrapper.

		Classified As	
		edible	poisonous
Class	edible	135	
	poisonous		91

(d) C4.5 wrapper.

		Classified As	
		edible	poisonous
Class	edible	131	4
	poisonous	4	87

(e) Contribution analysis.

Table 7.6: Summary of the *MUSHROOM* error rates for the test set.

7.4.3 The MONK's Dataset

The MONK's dataset was generated to compare the performance of different learning algorithms [162]. The problem describes an artificial robot domain, where three predicates using the following six attributes were used.

Attribute Number	Attribute Name	Attribute Values
1	`head_shape`	round, square, octagon
2	`body_shape`	round, square, octagon
3	`is_smiling`	yes, no
4	`holding`	sword, balloon, flag
5	`jacket_colour`	red, yellow, green, blue
6	`has_tie`	yes, no

The original neural network problem in [162] was to investigate binary classification where each attribute value was used as a binary input for a neural network. For the purposes of this work it was more useful to assign each attribute to an input of the neural network, with the attribute values assigned numbers between 0 and 1. A single output node was used to give a true or false classification.

7.4.3.1 MONK's Problem 1

The first problem (*MONKS1*) was to train a classifier to give a true (target output 0.9 for our purposes) or false value (target output 0.1) for the predicate

```
(head_shape = body_shape) or (jacket_colour = red)
```

Obviously for this problem the attributes `head_shape`, `body_shape` and `jacket_colour` are strongly relevant and the others are irrelevant.

The number of cases in each class were as follows.

	# Training Cases	# Stopping Cases	# Test Cases
true	62	108	108
false	62	108	108

The contributions for all of the attributes is shown in Figure 7.5. As expected the `head_shape`, `body_shape` and `jacket_colour` attributes all have quite large contributions in relation to the other attributes and so were retained for classification. These results confirm that the magnitude of the contribution is important.

These results also support the hypothesis that variation in contributions is not a key determinant in identifying the relevant attributes. Each of the strongly relevant attributes show a reasonable amount of variation in contribution.

Once again, wrapper attribute selection was also carried out for both C4.5 and neural network classification. The attributes selected for each of the three methods is summarised in Table 7.7.

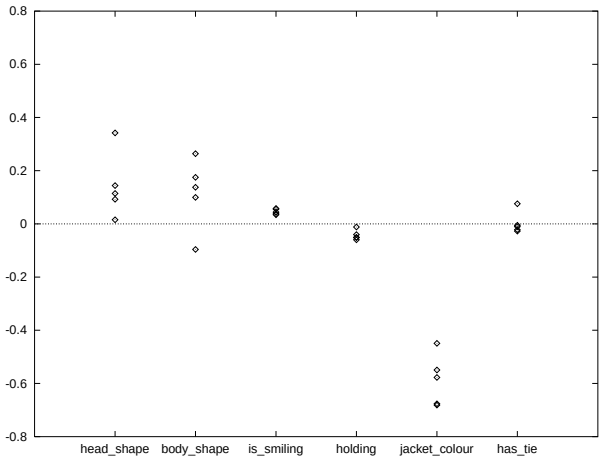


Figure 7.5: Contribution of each attribute for the *MONKS1* data.

Contribution	NN Wrapper	C4.5 Wrapper
`head_shape`	`head_shape`	`head_shape`
`body_shape`	`body_shape`	`body_shape`
`jacket_colour`	`is_smiling`	`jacket_colour`
	`jacket_colour`	

Table 7.7: Attributes selected for the *MONKS1* data.

Each of the attribute selection techniques chose the relevant attributes. The neural network with wrapper attribute selection included the additional attribute `is_smiling` because it did not increase the error rate. A different presentation order for the attributes removes this attribute from the chosen set.

All of the classifications achieved a 0% error rate on the test set, except for the neural network using all the attributes. The irrelevant attributes serve to confuse the true characteristics being represented in the dataset.

		Classified As	
		true	false
Class	true	91	17
	false	16	92

(a) Neural network with all attributes.

		Classified As	
		true	false
Class	true	108	
	false		108

(b) C4.5 with all attributes.

		Classified As	
		true	false
Class	true	108	
	false		108

(c) Neural network wrapper.

		Classified As	
		true	false
Class	true	108	
	false		108

(d) C4.5 wrapper.

		Classified As	
		true	false
Class	true	108	
	false		108

(e) Contribution analysis.

Table 7.8: Summary of the *MONKS1* error rates for the test set.

For this particular problem contribution analysis as the attribute selection technique is the superior method as the wrapper used with a neural network takes considerably more time for training. However, this is a very simple problem with obvious attribute choices.

7.4.3.2 MONK's Problem 2

The next problem (*MONKS2*) was to train a neural network to give a true or false value for exactly two of the following predicates

```
head_shape = round
body_shape = round
is_smiling = yes
holding = sword
jacket_colour = red
has_tie = yes
```

The key points to this classification are that:

- All of the attributes are relevant.
- This is an XOR problem that requires a network topology with two hidden layers for an optimal solution. As it is the contribution analysis that is being tested for a standard network topology, two hidden layers will not be used, with the understanding that the best possible accuracy will not be obtained.

The number of cases in each of the datasets was

	# Training Cases	# Stopping Cases	# Test Cases
true	64	71	71
false	105	145	145

If we look at the contribution for each of the attributes, as shown in Figure 7.6, we see that none of the attributes stands out as being more or less relevant than any other. This supports the hypothesis that it is not the variation in the magnitude of the contribution, but rather that each relevant attribute has a reasonably large contribution.

With the assumption that variation in contribution is not an indicator of attribute relevance all of the attributes were retained. The attributes chosen for each of the three attribute selection methods can be seen in Table 7.9. The four other

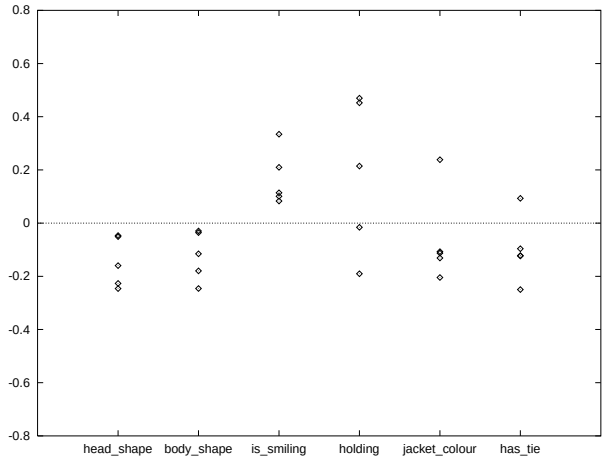


Figure 7.6: Contribution of each attribute for the *MONKS2* data.

Contribution	NN Wrapper	C4.5 Wrapper
`head_shape`	`head_shape`	`head_shape`
`body_shape`		
`is_smiling`		
`holding`		
`jacket_colour`		
`has_tie`		

Table 7.9: Attributes selected for the *MONKS2* data.

classifications were carried out as before, with no additional training runs done for the contribution attribute selection.

The error rates for all classifications are relatively high, as shown in Table 7.10. The wrapper technique chose only a single attribute for both the neural network and C4.5 classifiers, and the high error rates are due to a default classification of true being given. A result such as this is to be expected as all attributes are relevant and only one was used. Extending the search done by the wrapper may show an improvement in the attributes chosen.

For the classifications using all of the attributes true and false classifications are being differentiated to some extent. Contribution attribute selection is superior for this data set as a better set of attributes has been found than for the wrapper attribute selection, and with only one additional training run of a neural network. Obviously the choice of attributes using contributions is to some extent subjective, but is an excellent indicator of attribute relevance as this classification task

demonstrated.

		Classified As	
		true	false
Class	true	53	18
	false	34	111

(a) Neural network with all attributes.

		Classified As	
		true	false
Class	true	46	25
	false	17	128

(b) C4.5 with all attributes.

		Classified As	
		true	false
Class	true	71	
	false	145	

(c) Neural network wrapper.

		Classified As	
		true	false
Class	true	71	
	false	145	

(d) C4.5 wrapper.

		Classified As	
		true	false
Class	true	53	18
	false	34	111

(e) Contribution analysis (all attributes chosen).

Table 7.10: Summary of the *MONKS2* error rates for the test set.

7.4.3.3 MONK's Problem 3

The final MONK's (*MONKS3*) problem aims to give a true or false value for the predicate

(`jacket_colour = green and holding = sword`) or
(`jacket_colour` $\neq$ `blue and body_shape` $\neq$ `octagon`)

That is, `jacket_colour`, `holding` and `body_shape` are strongly relevant. This is a particularly difficult problem to solve as there are two conditions on `jacket_colour`. In addition to this 5% of the cases in the dataset are misclassified.

The number of cases in each dataset are

	# Training Cases	# Stopping Cases	# Test Cases
true	60	114	114
false	62	102	102

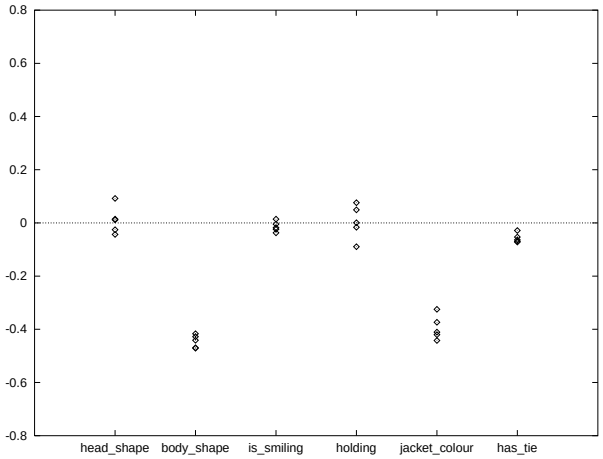


Figure 7.7: Contribution of each attribute for the *MONKS3* data.

From the contributions shown in Figure 7.7 the attributes `body_shape` and `jacket_colour` definitely contribute the most to the output of the network. The attributes `head_shape` and `holding` cluster around zero but have a large variation in contributions over the different networks, and so it was considered to be worth retaining them. The attributes chosen by the three attribute selection methods are summarised in Table 7.11.

It is worth noting that the training set did not contain any cases where

(`jacket_colour = green and holding = sword`)

was true. This meant that the choice of only `body_shape` and `jacket_colour` for the wrapper is to be expected.

Contribution	NN Wrapper	C4.5 Wrapper
`head_shape`	`body_shape`	`body_shape`
`body_shape`	`is_smiling`	`jacket_colour`
`jacket_colour`	`jacket_colour`	
`holding`		

Table 7.11: Attributes selected for the *MONKS3* data.

Overall the error rates for all classifications were surprisingly low (see Table 7.12). In particular, the error rate for the neural network and C4.5 dropped significantly after wrapper attribute selection and it is not clear why this should be the case. Further investigation of the properties of the test set will be required.

The error rate for the contribution attribute selection did not decrease, but neither did it increase. This is a useful result as it means that we have found the attributes that are relevant and reduced our classification times.

		Classified As	
		true	false
Class	true	86	28
	false	4	98

(a) Neural network with all attributes.

		Classified As	
		true	false
Class	true	109	5
	false	5	97

(b) C4.5 with all attributes.

		Classified As	
		true	false
Class	true	108	6
	false		102

(c) Neural network wrapper.

		Classified As	
		true	false
Class	true	108	6
	false		102

(d) C4.5 wrapper.

		Classified As	
		true	false
Class	true	88	26
	false	2	100

(e) Contribution analysis.

Table 7.12: Summary of the *MONKS3* error rates for the test set.

7.4.4 Solar Flare Dataset

The *FLARE* dataset gives the number of solar flares in a 24hr period and the conditions under which they occurred. The attributes used to describe solar flare activity are as follows:

Attribute Number	Attribute Name	Attribute Values
1	`modified_Zurich_class`	A, B, C, D, E, F, H
2	`largest_spot_size`	X, R, S, A, H, K
3	`spot_distribution`	X, O, I, C
4	`activity`	reduce, unchanged
5	`evolution`	decay, no_growth, growth
6	`previous_24hr_activity`	not as big as M1, one M1, more activity than one M1
7	`historically_complex`	yes, no
8	`became_historically_complex`	yes, no
9	`area`	small, large
10	`area_largest_spot`	$<= 5$, > 5

For each case the number of C (common), M (moderate) and X (severe) class flares were given. Ideally it would have been more interesting to identify cases where there were X-class flares, however only a total of 12 cases were available and so insufficient for the purposes of these experiments. So, the M-class flares were investigated.

The cases in the dataset used for this classification task (*MFLARE*) were given a classification of an M-class flare occurring (target output value 0.9) or an M-class flare not occurring (target output value 0.1). The number of flares were not used in this classification, it was simplified to be a true / false problem. The numbers of cases in each dataset were follows.

	# Training Cases	# Stopping Cases	# Test Cases
M-class flare occurred	34	17	17
no M-class flare occurred	145	73	73

The contributions of the attributes is reasonably high for most of the attributes, as shown in Figure 7.8. Only the attributes `activity`(4)
and `became_historically_complex`(8) contribute little to the classification. The contributions for these attributes cluster very close to zero and show little variation in the magnitude of the contributions over the different networks.

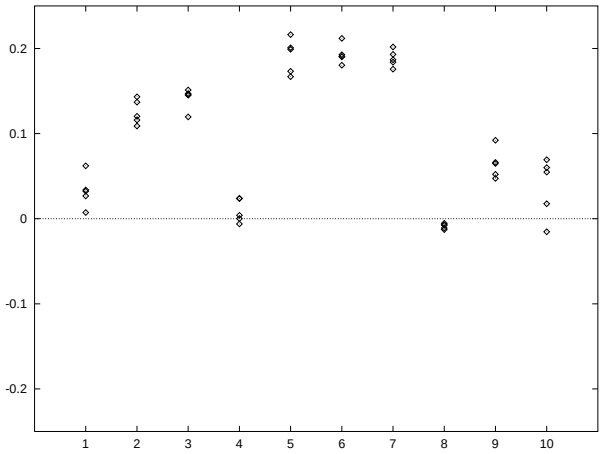


Figure 7.8: Contribution of each attribute for the *MFLARE* data.

The attributes selected for the three attribute selection approaches are summarised in Table 7.13.

Contribution	NN Wrapper	C4.5 Wrapper
`modified_Zurich_class`	`evolution`	`modified_Zurich_class`
`largest_spot_size`	`area`	`area`
`spot_distribution`		
`evolution`		
`previous_24hr_activity`		
`historically_complex`		
`area`		
`area_largest_spot`		

Table 7.13: Attributes selected for the *MFLARE* data.

The results of the five classifications on the test set can be seen in Table 7.14.

Using the wrapper with both the neural network and C4.5 there is no real change in the error rates when compared with the classification using all the data. In particular, using C4.5 with wrapper attribute selection finds only a small number of relevant attributes due to the bias in the data. Only 19% of the training

dataset are M-class flares. This means that a default classification of no M-class flare occurring gives an error rate of 19%, and any increase in the ability to identify M-class flares causes an overall decrease in the error rate. This overall decrease in the error rate causes the wrapper algorithm to stop searching prematurely.

The neural network trained on attributes selected using contributions produced similar results to the other neural network classifications. This means that we have correctly identified the relevant attributes for this classification task.

		Classified As	
		mflare	no mflare
Class	mflare	10	7
	no mflare	8	65

(a) Neural network with all attributes.

		Classified As	
		mflare	no mflare
Class	mflare	3	14
	no mflare	3	70

(b) C4.5 with all attributes.

		Classified As	
		mflare	no mflare
Class	mflare	10	7
	no mflare	8	65

(c) Neural network wrapper.

		Classified As	
		mflare	no mflare
Class	mflare	5	12
	no mflare	4	69

(d) C4.5 wrapper.

		Classified As	
		mflare	no mflare
Class	mflare	11	6
	no mflare	9	64

(e) Contribution analysis.

Table 7.14: Summary of the *MFLARE* error rates for the test set.

7.4.5 Variations in Contribution

In both the *MONKS3* and *FLARE* datasets we saw indications that it is the magnitude of the contributions, not the variations in the magnitude of the contribution, that determine the relevance of an attribute. To investigate this further, an additional classification was carried out for the *MFLARE* dataset.

For the neural network trained on the *MFLARE* dataset, attributes that had a variable contribution
(`modified_Zurich_class`(1), `area`(9) and `area_largest_spot`(10)) were also removed. That is, only the attributes `largest_spot_size`(2), `spot_distribution`(3), `evolution`(5), `previous_24hr_activity`(6) and `historically_complex`(7) were used in the classification.

For this classification the accuracy decreased from 83% for the original contribution attribute selection to 55% for the test set. This indicates that the additional attributes are, at least, weakly relevant attributes and should not have been removed. That is, it is the magnitude of the contribution that is important not the variation in the contributions for a given attribute. Only attributes that have a contribution clustering close to zero should be removed from the set of attributes.

It must be noted, once again, that attribute selection using contributions is to some extent a subjective choice, which may be harder to make on less well behaved real-world datasets. However, the contributions can at least provide an indication of the relevance of an attribute for a given classification task.

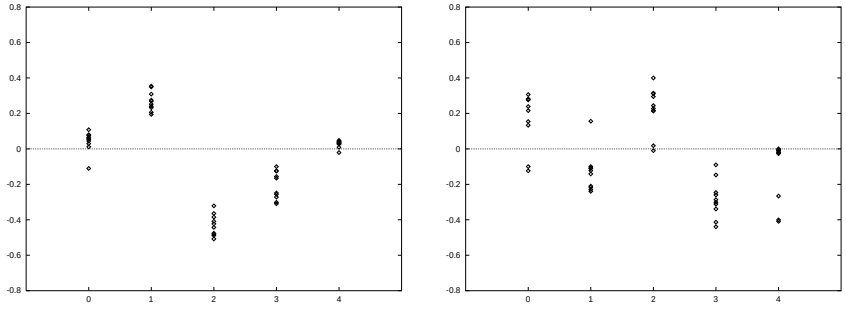
7.5 The Effects of Noise on Contributions

Each of the datasets mentioned previously are fairly well behaved in that there is minimal noise in each dataset, the attributes used have been chosen by domain experts for their information content and because they provide a reasonably clear separation between each of the classes in the attribute space Generally, there will be little or no noise in the actual attribute values and so each of these datasets can be used for investigating the effects of noise in a dataset.

7.5.1 Irrelevant Attributes

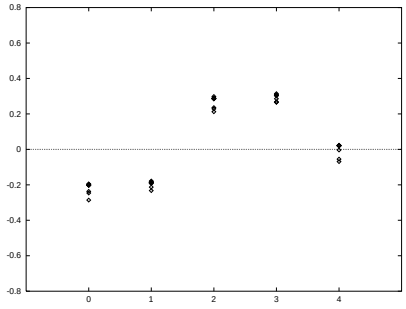
For each of the datasets discussed previously noise was added firstly by adding an additional attribute that was made up of random numbers, then the contribution of all the attributes was determined. The contributions of the original attributes and the extra noise attribute can be seen in Figure 7.9. The last attribute in each plot is the noise attribute.

In the majority of cases, the contribution of the noise attribute clusters fairly close to zero. That is, an irrelevant attribute is likely to be identified by its almost zero contribution. Once again, this supports the hypothesis that it is the magnitude of the contribution not the variations in the contributions that determine an attributes relevance. However, the ability to identify irrelevant attributes in this way will depend on the specific characteristics of a given dataset.

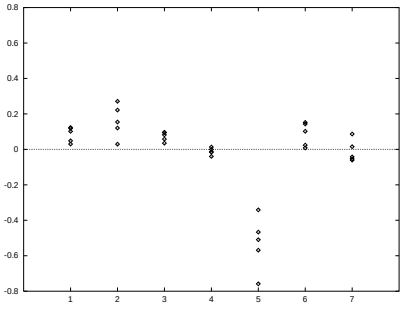


(a) SETOSA.

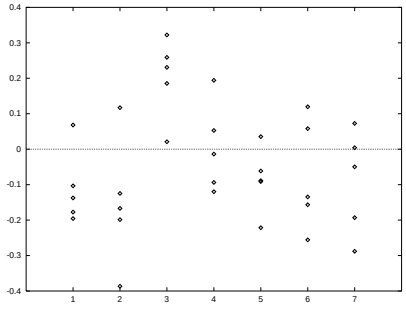
(b) VERSICOLOR.



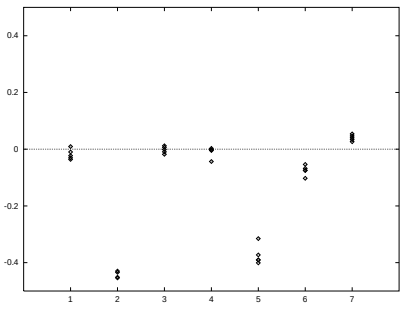
(c) VIRGINICA.



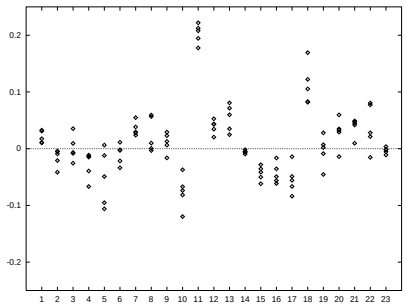
(d) MONKS1.



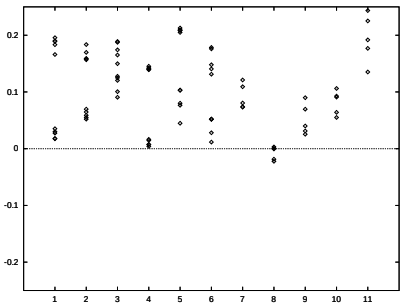
(e) MONKS2.



(f) MONKS3.



(g) MUSHROOM.



(h) MFLARE.

Figure 7.9: Contribution for each dataset with an additional noise attribute added. In each case the noise attribute is the one on the far right.

7.5.2 The Effects of Noise on Attribute Contributions

A specific attribute may be identified by an expert as a strongly relevant attribute. However, if the attribute values are incorrect or contain a lot of noise the attribute will be irrelevant for the purposes of a classification task.

The effect of adding increasing amounts of noise on the magnitude of the contribution of a strongly relevant attribute was investigated. The list of datasets chosen and the strongly relevant attribute used are listed below. These attributes were chosen as they had large contributions in the previous experiments.

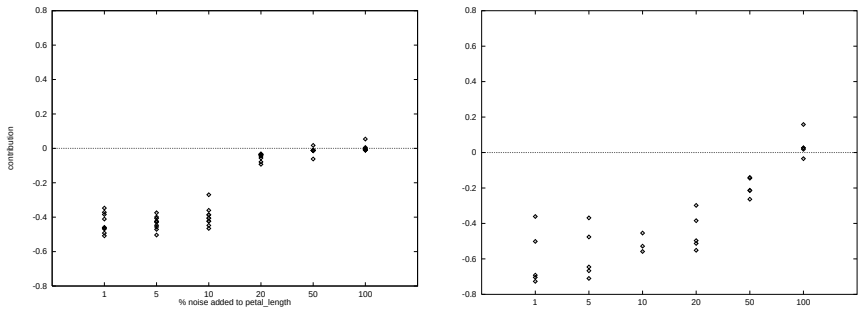
Dataset	Relevant Attribute
SETOSA	`petal-length`
MONKS1	`jacket_colour`
MONKS2	`holding`
MONKS3	`jacket_colour`
MUSHROOM	`stalk-root`
MFLARE	`historically_complex`

Noise levels of 1%, 5%, 10%, 20%, 50% and 100% were added to the attribute values for the given attribute to generate six new data sets. The networks were retrained for the new data as before, and the contributions for the strongly relevant attribute calculated.

Figure 7.10 shows just the contributions for the strongly relevant attribute with increasing levels of noise. The contributions of the remaining attributes changed a little to adjust for the loss of information in the given attribute, but generally remained the same.

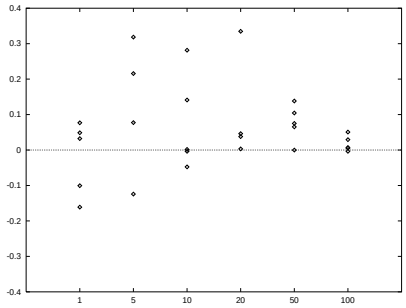
Clearly in each case the contribution of the attribute decreases towards zero. In general, each seems to show a decrease in the variation of the contributions over the five networks as well. Again, this supports the idea that it is not the variation in contributions over different networks that determines the relevance of an attribute, rather, it is the magnitude of the contribution relative to the other attributes that is important.

As expected, if the noise in an attribute is high it will be of little use in a classification and can be removed from the attribute set. In terms of the neural network a contribution that is "close to zero" indicates a noisy or irrelevant attribute.

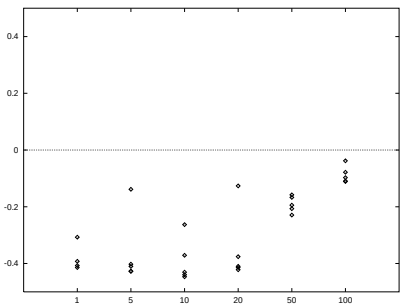


(a) SETOSA.

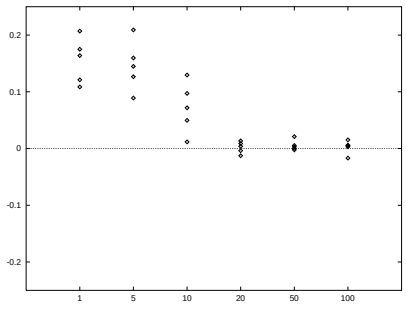
(b) MONKS1.



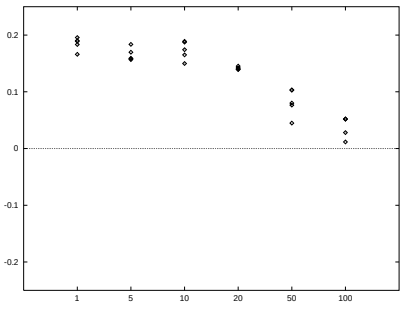
(c) MONKS2.



(d) MONKS3.



(e) MUSHROOM.



(f) MFLARE.

Figure 7.10: Contribution for a strongly relevant attribute for each dataset. (% noise added vs contribution)

7.6 Application to Remotely Sensed Data

Contribution analysis is clearly useful when used on datasets with small numbers of attributes. How well does it perform when used on datasets with large numbers of noisy or irrelevant attributes?

Contribution analysis was applied to remotely sensed data, where the classification task was to distinguish between two classes, grass and trees, using 151 attributes. A discussion of this was also published in Milne [111].

Classification using all 151 attributes resulted in no distinction being made between the two classes. The average error rate on the test set was 38%, but varied from 24% to 53% for individual networks. This indicates that the neural networks were not able to distinguish between relevant and irrelevant attributes, and were unable to find the central characteristics that distinguish the two classes.

Attribute selection was carried out on the 151 attributes using contribution analysis. The magnitude of the contributions varied from around -0.04 to 0.03 and so thresholds for selecting attributes of ± 0.01, ± 0.02 and ± 0.03 were tested. That is, if attributes fell within the given range they were removed from the training data.

Neural networks were then trained on the attributes selected. The error rates for these classifications on the test set can be seen in Table 7.15.

Dataset	# Attributes	Av. Test Error
all attributes	151	38%
attributes with contribution >0.01 and <-0.01	53	5%
attributes with contribution >0.02 and <-0.02	12	7%
attributes with contribution >0.03 and <-0.03	3	14%

Table 7.15: Average error rates for classification of remotely sensed data as reported in [111].

In all cases, after attribute selection the error rates decreased significantly. In addition, the error rates for the different neural networks trained on the same training dataset only differed from the average by at most a few percent between individual neural networks. That is, we are able to obtain more reliable and consistent classifications from neural networks if we remove irrelevant and noisy data.

The classifiers were then used to classify the entire remotely sensed image on which the training dataset was based. In spite of the low error rates, it was found that the classification using the threshold of 0.01 still had too many attributes as there was little differentiation of the two classes over the entire image. On the other hand, the classification with the threshold of 0.03 had too few attributes, and again, was unable to show any differentiation between the two classes over the entire image.

Overall it was demonstrated that it is possible to use contribution analysis for attribute selection with large number of attributes. A number of training runs may be required to determine which is the best subset of attributes, however, it is significantly less time consuming than using wrapper attribute selection with neural networks.

7.7 Discussion

This chapter discussed a new technique for selecting relevant attributes for a number of different datasets when using neural network classifiers. A comparison between the wrapper method and contribution analysis for attribute selection on neural networks was carried out for well understood data sets. Attribute selection for large numbers of attributes was also discussed.

Only the contributions of inputs to a neural network with a single output node were considered here. Further work is needed to investigate the best approach for networks with more than one output node.

Selection of attributes using their contributions was done by the author based on the absolute magnitude of a given contribution and the relative magnitudes of the contributions for each of the attributes. For the remotely sensed data used in this work, extensive experimentation showed that a threshold of ± 0.2 was an appropriate choice. Further work will need to be done to determine if this is a generally applicable threshold for the relevance of an attribute from its contribution.

It must be noted that both the wrapper method and contribution analysis are heuristic search techniques and can only choose a good set of attributes, not necessarily the optimal set of attributes. However, attribute selection using contribution analysis is still a considerably faster approach for neural networks and overall performed no worse than the wrapper method.

It is not a requirement that the error rates of a classifier must decrease after attribute selection. By reducing the number of attributes we are at least reducing the time taken to generate a classification. In some cases the error rates of the classifiers after attribute selection actually increased than when using all attributes. This is due to relevant attributes not being chosen or irrelevant attributes retained. There is no reason why a number of iterations of attribute selection can't be carried out to find a better set of attributes, and so at least maintain the accuracy.

Overall, when using contribution analysis for attribute selection only those attributes with a contribution that cluster close to zero should be removed. However, as the datasets contain increasing numbers of attributes the magnitude of the contributions will decrease. The relative contribution is still the important

factor in determining relevance.

7.8 Conclusions

We now have two useful attribute selection techniques, the wrapper method and contribution analysis specifically for use with neural networks. This allows us to generate a large number of attributes, as discussed in Chapter 6, and then remove the irrelevant information. This is of particular use in the context of this work as many of the attributes generated will be irrelevant for a given classification task.

The application of these attribute selection techniques will be used further in later chapters. Next we look at other ways of improving classification accuracy.

Chapter 8

Improving Classification Accuracy

So far we have discussed methods for extracting the most useful information from remotely sensed data, but this only addresses part of the problem. Additional problems include:

Misclassified training data. Errors in survey data due to problems such as inexact identification of location, differing expert opinion, and vague concepts[1].

Incomplete class information. It is not necessarily desirable or even possible to enumerate all classes that could occur in a remotely sensed image, and harder again to collect training data for them. A specific example of this is mapping forest types. It is difficult, if not impossible, to carry out sufficient surveys to obtain data on each possible forest type, for a given area.

Small training datasets. In this domain it is difficult and expensive to generate large training datasets. Data collection is expensive as it typically requires ground surveys, over large areas.

Misclassifications from classifiers. Misclassification of unseen data is, in part, due to misclassified training data and incomplete class information. It is also

[1] An example of this is the forest types as defined by the NSW Forestry Commission [57]. The forest types define proportions of particular species that occur in association. The proportions are defined in terms of ranges making this is a highly subjective classification.

due to the way in which the classifier generates a model of the training data.

There are, however, a number of techniques we can use to reduce the number of misclassifications.

Other work, such as that discussed in [93, 76, 70], has shown that binary classification requires fewer attributes and produces more accurate classifications due to the simplification of the task. Initial experiments carried out as part of this work published in [113] confirmed these results.

Misclassifications can also be due to the characteristics of remotely sensed data. For example, the spectral signatures for bare soil or rocks can be similar to that of man made structures [102]. This can be due to the small number of broad spectral bands being used, however, Price [130] found that even using a large number of narrow spectral bands, separation of some classes is still not possible. Problems related to the limitations of the available data should not stop us from using the data that is available to generate a reasonable classification.

Pixel unmixing has been used to avoid misclassification of remotely sensed data (for examples see [102, 58]). The disadvantage of this technique is that data samples are needed for pure classes. That is, the spectral characteristics for just soil or just vegetation are needed. To complicate matters further, spectral signatures can vary quite significantly with varying conditions, and so pure class data must be collected under a variety of conditions.

Stone et al [154] developed a land cover map of South America using AVHRR data. Classifications using vegetation indices were possible with an overall error rate of around 10%. However, a problem in this study was the misclassification of sites in the study area. It was estimated that 8.5% of the map was between 76% and 89% reliable, while 6.5% of the map was less than 75% reliable. While the use of low resolution data is limiting, its use should not result in such varying classification reliability.

Given that misclassification errors can arise, and may be for a number of reasons, it would be preferable to only classify areas that are sufficiently similar to classes that are understood and not try to classify anything else. That is, only give a classification when a number of techniques are in agreement and not have to make the assumption that we have data on all possible classes in an image.

In this chapter we investigate a number of techniques that will help us to increase classification accuracy for noisy, small or incomplete training sets.

8.1 Training Dataset

In this chapter we look at a simple classification of the *CSU* dataset. Four broad classes were identified as the main classes in the image (`grass`, `trees`, `urban` and `water`). A small dataset for a simple classification task was deliberately chosen so that the improvements demonstrated could be attributed to the technique, not the vagaries of the data.

A multi-class neural network was trained to recognise each the four classes, using only the spectral data as inputs. The inputs were scaled to values between 0 and 1. The network topology used was four inputs, two hidden nodes and two output nodes. Each of the classes was mapped to target output values as shown below. A total of 433 training cases were extracted from the image and were split up into three sets, as described in Section 3.3.

Class	Target Output Values	Total Cases	Training Cases	Stop Cases	Test Cases
`grass`	0.1 0.1	109	56	29	24
`trees`	0.1 0.9	103	65	22	16
`urban`	0.9 0.1	103	64	16	23
`water`	0.9 0.9	118	74	20	24

The multi-class classification gave an error rate of 4.6% on the test set for the four class neural network (see Table 8.1), and we can generate a reasonably accurate map from the image (see Figure 8.1). Comparison should be made with the labelled image in Figure 3.1.

In Figure 8.1 we can see that a few of the lakes in the image have been misclassified as `trees`, which is not unreasonable considering that most are covered by water weeds. Boundaries around some of the buildings have been given a classification of `water`. As well, in the lower right quadrant some of the buildings and trees have been confused. This is also a reasonable error as many of the buildings have overhanging trees.

		Classified As			
		grass	trees	urban	water
Class	grass	21	2	1	
	trees		16		
	urban		1	22	
	water				24

Table 8.1: Classification accuracy on the test set for the best multi-class neural network.

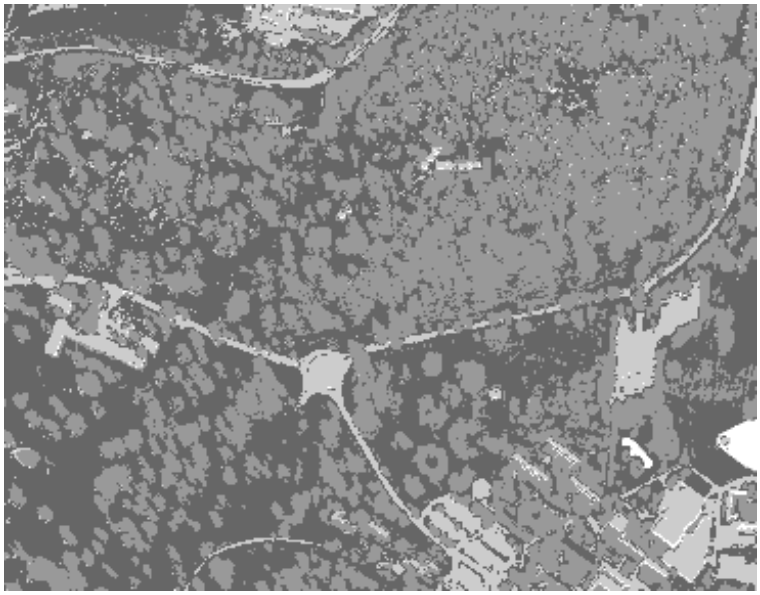


Figure 8.1: A four class classification of the image using a multi-class neural network.

It is important to note that data specific properties, such as the effects of shadow and mixed pixels, were not investigated as part of this work. However, the classification framework developed in this thesis lends itself well to this kind of investigation.

8.1.1 Incomplete Information

In practice a reasonably high accuracy classification in not necessarily going to be possible. The classification in Figure 8.1 had complete class information and relatively accurate training data. That is, the training data contained classes

that cover all objects in the image, and the cases for each class were reasonably representative of the characteristics for the given class.

In classifications where this level of detail is used high accuracies are readily obtained. However, when we start to consider more detailed classifications, such as identifying specific plant species, it is much harder to generate high accuracy classifications. This is, in part, due to the difficulty in collecting enough training data of sufficient quality.

Even for broad class classifications as done here, errors will still occur. If we assume that it is better to give no classification to a pixel rather than an incorrect classification we can still produce reasonably accurate classifications.

To test this the `grass` class was left out of the training set and stopping set for the dataset mentioned in the previous section. A multi-class neural network, with the same topology was trained on the data for the `trees`, `water` and `urban` classes only. The networks still have two outputs, and the output 0.1 0.1 is given an `unknown` classification.

The error rate on the test set has apparently dropped from 4.6% to 1.1%, as seen in Table 8.2. The neural network was then applied to the 24 `grass` pixels from the original test set, and 15 were misclassified as `trees` and 7 misclassified as `urban`. Two of the grass pixels have been given the `unknown` class.

		Classified As		
		trees	urban	water
Class	trees	16		
	urban	1	22	
	water			24

Table 8.2: Classification accuracy of a neural network.

As would be expected the classification of the entire image is not as good as the original four class classification, as seen in Figure 8.2. The classifier was not trained to recognise `grass` pixels and so pixels that should have this class have been misclassified as belonging to one of the other classes. On the test set, the `grass` pixels have been classified as `trees` or `urban`. A few of the pixels have been given the `unknown` classification, though this will not always happen in practice and depends on the number of classes and the network topology.

In this particular case the pixels given the `unknown` classification are largely bound-

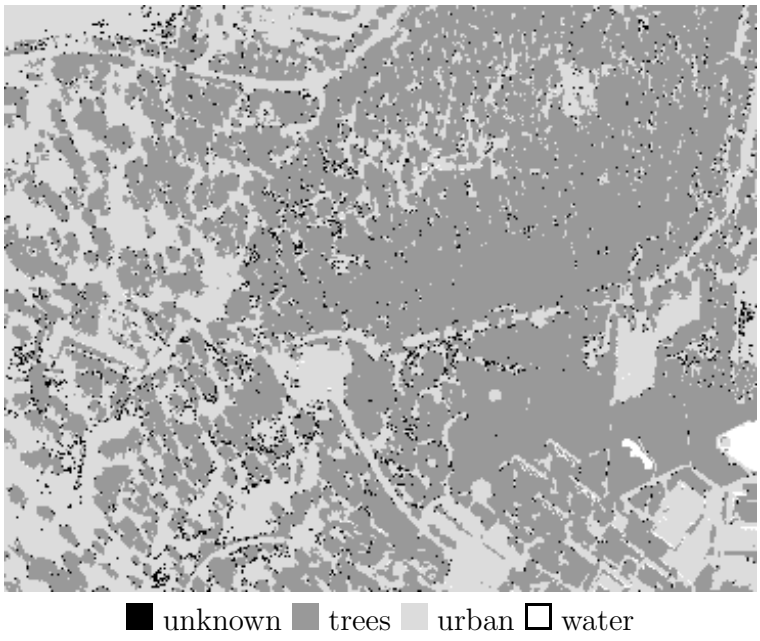


Figure 8.2: Neural network map, `grass` class left out.

ary pixels. These pixels can contain spectral information from one or more classes and so are likely to be misclassified. For the ABVS imagery in particular there is also a slight offset that occurs between the bands during image acquisition[2] that will also cause misclassification of pixels at the boundaries of classes.

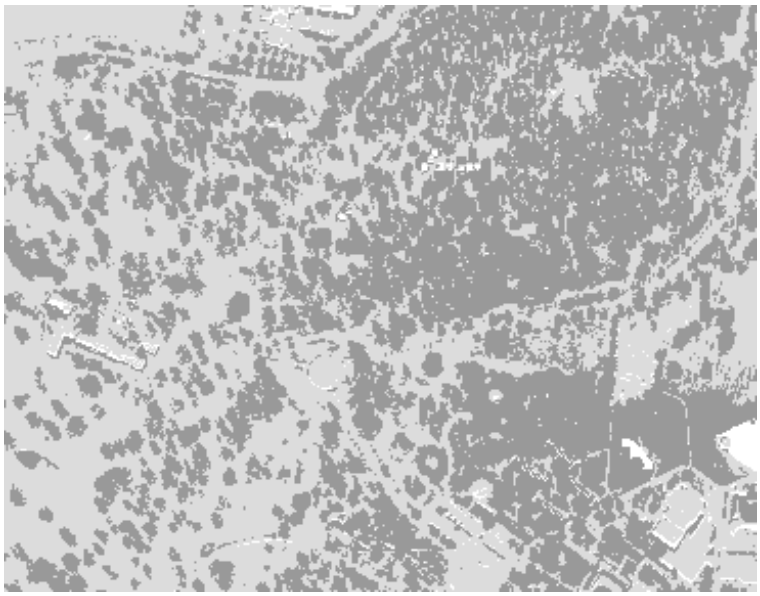
The misclassified region at class boundaries is generally one to two pixels wide in this specific case, but this will vary from image to image. A more detailed investigation of the effects of boundary pixels was not carried out as part of this work.

The poor performance demonstrated in classifying unseen cases using neural networks is not due the classification technique or the configuration of the networks used. Similar results are obtained for other classification systems.

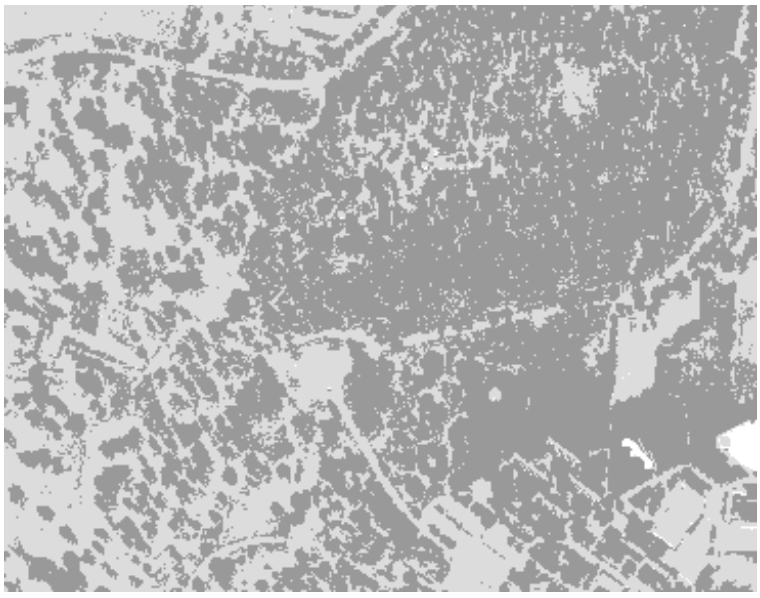
Two further classifications were generated using C4.5 and IBL(k=1). The same data that was used to train the neural networks were used to train the additional classifiers. Both gave exactly the same error as the multi-class neural network

[2]Rectification of the ABVS image has not been carried out and is a preferable solution. However, as the ABVS is an experimental system, rectification of images is being investigated by the researchers at the Spatial Analysis Unit at Charles Sturt University, Wagga Wagga, Australia.

classification – an overall error on the test set of 1.1% as shown in Table 8.2. As expected, the low error rates on the test data are not reflected in the classifications of the entire image (see Figure 8.3) due to incomplete information.



(a) C4.5 classification.



(b) IBL(k=1) classification.

trees urban water

Figure 8.3: Multi-class classifications.

The classifications given by the three classifier systems for the incomplete training data have very low error rates on the test set. However, when we apply the classifiers to an entire image the error is apparently quite high when visually evaluated. While we are unable to quantify the error over the entire image as we only know the classes of a relatively small number of pixels, the error is high enough that qualitative assessment can be used.

A high error rate over the entire image is, of course, expected. We know that the classifier will never be able to recognise `grass` as it has not been trained to do so. When using all of the available data to train the neural network we are able to produce a reasonable classification of the entire image.

Within a vegetation mapping domain noisy and incomplete data is always a possibility. Misclassifying training data is possible for a variety of reasons, as with any other domain. This problem is compounded when we try to carry out more detailed classifications that may not contain training data for all possible classes.

Overall we have the following problems:

- If there are classes that we do not have training data for our classifiers will never be able to recognise them.
- Misclassifications can occur when there are noise or errors in the training data.

We will now look at techniques that will help to reduce the number of misclassifications for datasets that are incomplete or noisy, using the three class dataset as described here.

8.2 Thresholding Neural Networks

A major problem with misclassification of pixels arises from the fact that classifiers pigeon hole all cases into one of the known classes. A neural network, however, produces continuous outputs, typically in the range 0 to 1. The target outputs are at specific points within this range. The actual outputs rarely fall exactly at these points and so the output values are partitioned to give a range of outputs for each of the target values. Input cases are assigned the class that corresponds to the range in which the output value falls.

For the networks described in the previous section, to get a classification the output range of each output node was divided into two. That is, an output > 0.5 meaning the node has given a classification of 0.9, otherwise the classification is 0.1. These values for both outputs of the neural network are then mapped to one of the four classes including the `unknown` class. However, if an output is close to 0.5 which target output value should it be mapped to? Rather than simply dividing the output space of a neural network between the known classes we can instead threshold the outputs and only accept a classification when it is "close" to an expected output value. The results of this work was also discussed in [113].

Using a number of networks with different input data, a range of thresholds were trialled. It was found that for networks with two target output values per node a threshold of $\pm 10\%$ of the target output values minimised the numbers of misclassifications arising from noisy or incomplete data, and maximised the numbers of correct classifications. So, in the case of the networks described in the previous section, a class is given only when the output is within ± 0.1 of the target output values. When an output falls outside of this range the case is given an `unknown` classification. This is not an error, as such, it just signifies that the pixel being classified is not similar enough to the classes that we have data for.

If we do this for the network trained on three classes we can still obtain an overall error rate of 3.4%, as shown in Table 8.3.

When we apply the thresholded neural network to the `grass` pixels from the original test set, 11 are misclassified as `trees` and 4 as `urban`. However, 9 of the `grass` pixels have been correctly given an `unknown` classification. This is a significant improvement over the neural network trained on just the three classes.

If we look at the map produced from the thresholded neural network we again

		Classified As			
		unknown	trees	urban	water
Class	trees		16		
	urban	2	1	20	
	water				24

Table 8.3: Classification accuracy of neural networks after thresholding the output values.

see a significant improvement (see Figure 8.4). A large proportion of the `grass` pixels have been given an `unknown` classification.

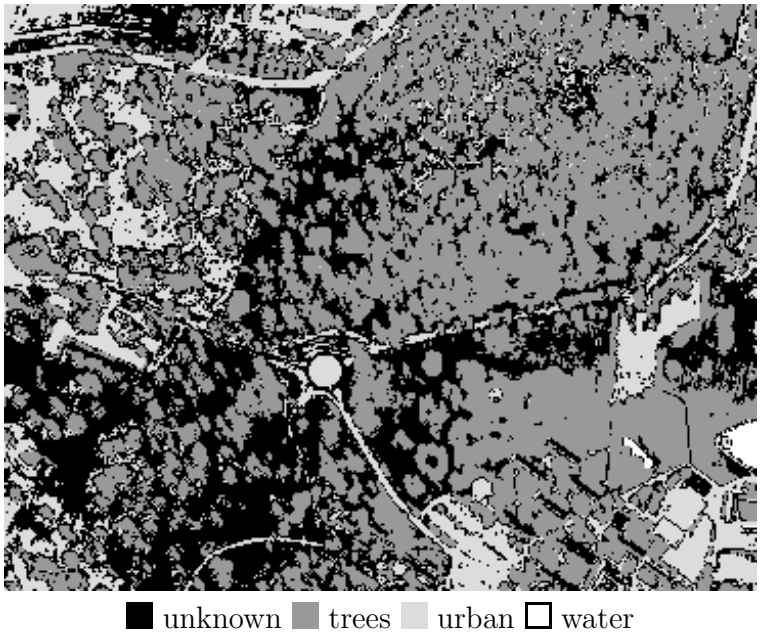


Figure 8.4: Neural network map with thresholding applied to the output values.

Although we have been able to limit the classification to pixels that are close to cases we have seen before there are still problems. In the thresholded network map there are still a large number of misclassifications. In the top left hand quadrant of the image there are `grass` pixels that have been misclassified as `urban` and in the bottom left quadrant as `trees`. Only two of the lakes have been identified.

Thresholding the outputs of a neural network can improve classification accuracy, but will only work for this type of neural network classification. We now look at ways to improve classification accuracy when using any classification system.

8.3 Classifier Configuration and Training

The increasing complexity of a classification task can result in increasing error rates. Classification problems with large numbers of attributes and classes will tend to give higher error rates. As discussed in Chapter 6 irrelevant attributes will also result in errors in classification. Similarly, a large number of classes requires a more complex division of the attribute space by the classifier system and so can increase the error rate.

Murre [115] described a modular hierarchical neural network methodology that more accurately reflects the structure of the human brain. It consisted of small modules that solve specific problems and are then combined in a hierarchical fashion. Smaller interacting networks are not only much faster to train, but also require smaller numbers of inputs to achieve a given task.

The use of a binary tree structure to simplify the classification process, was demonstrated in [68, 76]. A case is classified through a chain of binary classifiers, similar to a decision tree. Each node in the binary tree is a binary classifier that is trained to split the decision between two alternatives. This approach reduces the number of attributes required for each classifier and the accuracy of the resulting classification increased.

Binary classifiers can also be trained to distinguish one class from all other classes [93]. As with the *IRIS* and *MFLARE* experiments in Chapter 7, the target classes for the training data are grouped into two classes – in a single specified class or not in that class.

By breaking down a large classification problem into a series of smaller ones we can improve the accuracy of individual classifiers, and so the reliability of the maps generated. As we shall see, an additional benefit is that a different topology forces a different view of the data and so enables the number of incorrectly classified pixels to be reduced.

8.3.1 Binary Classifiers

Binary classification is achieved by grouping the training cases into two classes – the single class that is to be distinguished and the remainder of the cases from the other classes grouped into a single class. Training a classifier in this way finds

the characteristics of the most consistent class.

The topology of the binary neural networks used for this work are as follows. A binary network has a single output. The class to be recognised is given a target of 0.9 and the remainder of the classes are given a target of 0.1. The outputs of the network are also thresholded as discussed in Section 8.2.

Binary neural networks were trained to recognise each of the classes `trees`, `urban` and `water`. The error rates for these neural networks are similar to the multi-class neural network (see Table 8.4).

This approach gives us two measures of error. Firstly, the overall error gives the number of cases that have not been assigned their correct class. Secondly, the misclassification error is the number of cases that have been given an erroneous classification, which does not include the cases given an `unknown` classification. The cases given an `unknown` classification are not considered to be in error, rather are those that require further investigation.

When measured this way the overall error – all pixels that can not be correctly or reasonably accurately classified – for the `tree` class, for example, is 4.7% while the misclassification error – the pixels that have been given an incorrect classification – is 1.6%.

It was also found that the variance between the outputs of the neural networks trained on the same data but with different initial weights is considerably smaller for binary networks than for multi-class networks. This means that a pixel being classified by a number of different, trained binary classifiers all give outputs for a specific input within a much smaller range of output values. Simplifying the problem to be a binary classification has meant more consistent classifications can be obtained and so the reduced flexibility of the configuration is not a problem.

The classifications produced by the individual binary neural networks can be seen in Figure 8.5.

As can be seen in Figure 8.5(a) there has been some overestimation of the `tree` class – mostly grassed areas. However, the tree canopies are clearly visible.

The `urban` classification has a large amount of soil and rock areas included in it. But, the urban features have been clearly identified.

The `water` classification has minimal misclassifications. However, three of the

		Classified As		
		trees	not trees	unknown
Class	trees	16		
	not trees	1	44	2

(a) Trees.

		Classified As		
		urban	not urban	unknown
Class	urban	20		3
	not urban		40	

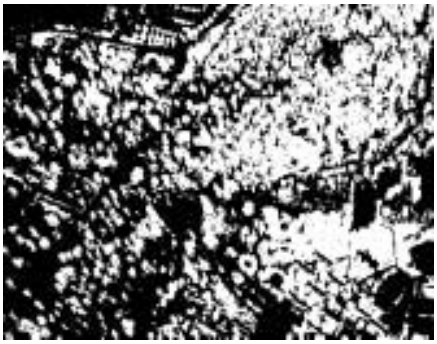
(b) Urban.

		Classified As		
		water	not water	unknown
Class	water	24		
	not water		38	1

(c) Water.

Table 8.4: Classification using binary neural networks.

lakes have not been identified, which is due to a thick coverage of water weeds. Again, this is an anomoly that justifies an `unknown` classification and would warrant further investigation for mapping purposes.



(a) Trees.

(b) Urban.

(c) Water.

■ not in the given class □ in the given class

Figure 8.5: Maps from binary neural networks classifiers.

For comparison, binary classifications using C4.5 and IBL were also generated.

The results on the test set for the C4.5 classification can be seen in Table 8.5 and the classifications over the entire image can be seen in Figure 8.6.

The C4.5 classification was better at identifying urban features. All grassed and soil areas have been given a `tree` classification (see Figure 8.6(a)). The `water` class has been substantially overestimated.

		Classified As	
		trees	not trees
Class	trees	16	
	not trees	1	46

(a) Trees.

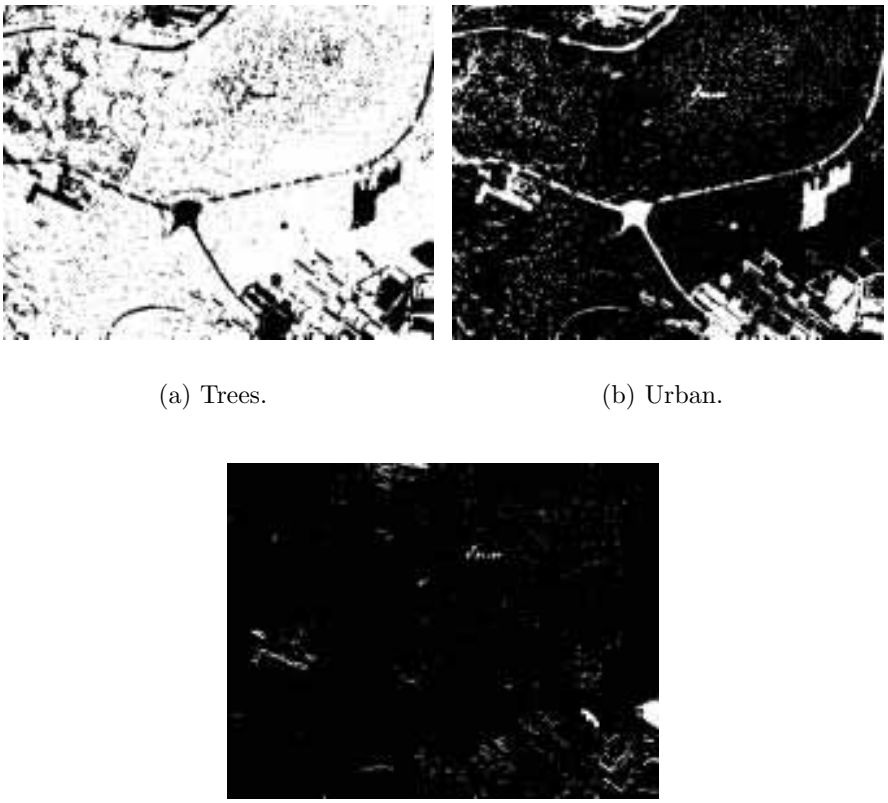
		Classified As	
		urban	not urban
Class	urban	22	1
	not urban		40

(b) Urban.

		Classified As	
		water	not water
Class	water	24	
	not water		39

(c) Water.

Table 8.5: Classification using binary C4.5 classifiers.



(a) Trees. (b) Urban.

(c) Water.

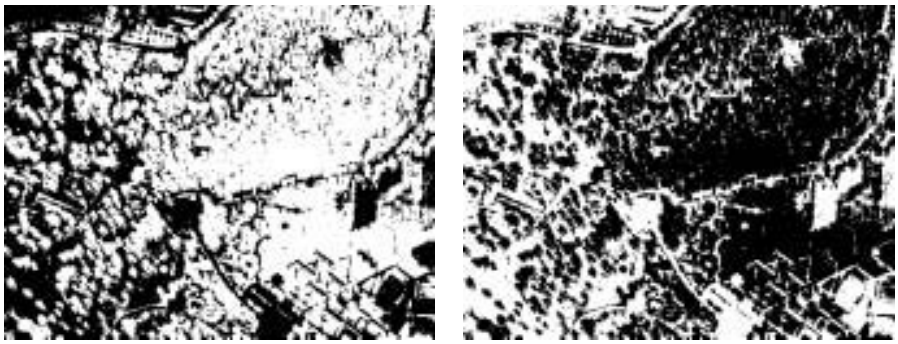
■ not in the given class □ in the given class

Figure 8.6: Maps from the binary C4.5 classification.

The IBL classification used the three nearest neighbours. If the IBL classification used only one nearest neighbour the result should not be significantly different to a multi-class classification. The error on the test set is identical to the binary error for C4.5, as shown in Table 8.5. The classifications for the entire image can be seen in Figure 8.7.

In this case, the `tree` and `urban` classes have been overestimated, while the `water` class has only a relatively small overestimate for class membership.

All three binary classifier systems have shown improvement in the classification accuracy, even though `grass` pixels are still being misclassified. We also see, particularly in the case of the IBL classification, that the occurrence of the `water` class has been over-estimated. As we need to combine binary classifications we



(a) Trees. (b) Urban.

(c) Water.

■ not in the given class □ in the given class

Figure 8.7: Maps from the binary IBL(k=3) classification.

can do this in such a way that reduces the overall error rate even further.

8.4 Multi-Strategy Classification

Real-world problems rarely satisfy all the requirements of a single strategy classification technique, it is more common for problems to only partially satisfy classifier pre-conditions [160]. Specifically in the case of neural networks, Thiria et al [161] state that the complexity of remotely sensed domains mean that a single classifier approach is not possible. Even if a dataset is a correct and complete description of a given domain it may be too complex to use with learning algorithms [141].

An alternative approach is to integrate a number of different strategies in some way. This can be done in one of two basic ways.

- The output of one classifier is used as input to a further classifier or arbitration scheme.
- Completely new classifiers can be created by merging the algorithms from existing classification techniques in some way.

The first approach is the one being considered in this thesis as there already exist a large number of excellent classification algorithms that have been proven to be effective. The central problem being addressed here is the amount and quality of the training data, not the issues with the classification algorithms themselves.

Layered neural networks were used by Yoshida and Omatu [179] to classify Landsat TM data. They were able to produce more realistic classifications as compared with standard backpropagation networks and maximum likelihood classification. The layered approach firstly classified pixels in the image into broad categories using a Kohonen self-organising map. Training data is then selected from geographical information and the self-organising feature map, which is used to train a backpropagation network. The results from this network are further improved by deleting pixels that are incorrectly classified from the training data set and a further backpropagation network trained.

The work of Yoshida and Omatu found that the mean squares error of their layered approach was not smaller than a standard backpropagation network. However, the layered approach was considered an improvement as it does not require complete class information and is more resistant to noise in the data. The layered approach increased the overall accuracy to 85%, from 61% for both standard backpropagation network and maximum likelihood classifiers.

Van Allen et al [4] used a combined approach to identify objects in noisy images. Firstly a nearest neighbour neural network classifier was used to extract common features from object shapes with reinforcement learning rules used to store the extracted features. Next the boundaries of the objects were completed using the Boundary Contour System [166].

In this case new classes could be identified for unfamiliar objects, mitigating the effects of incomplete information. A strength of this approach was considered to be the flexible system architecture provided by decomposing the problem into several stages. This modular approach also meant that each stage of classification could be individually optimised and so improve classification accuracy.

Thiria et al [161] found that the requirement of significant amount of a priori knowledge about a classification task made determining neural network topologies difficult. Their belief is that complex problems can not be solved with a single neural network, no matter how sophisticated it is. Their approach was instead to use a number of simple back-propagation neural networks that co-operate together.

Thiria et al [161] built a classification system of co-operating neural networks – the output of one or more of the back-propagation neural networks is used as input a successive layer of one or more neural networks. Each network in the architecture is dedicated to a specific task and is used to perform successive processing of the data. This approach meant that changes to the data or the problem to be solved could be more easily accommodated by adjusting the system architecture, rather than re-training the neural networks. This also means that errors in one module have the chance of being corrected in others.

Similarly Rogova [136] combined the results of multiple neural networks using the Dempster-Shafer theory of Evidence[3] for character recognition. Classification improvements of 15-30% were obtained as compared with the best single classifier.

Shavlik and Towell [141] combined the use of explanation based learning (techniques such as C4.5) and neural networks to recognise a variety of objects, such as chairs and cups. The advantage of the explanation based systems are that they only require a small number of examples to learn a concept, however, they do not function well with uncertainty, incomplete or changing information. Neural net-

[3]Used to assign probabilities to classifications and so determine the accuracy.

works, on the other hand, require large amounts of training data, training times are significantly longer and are difficult to interpret, but are better able to handle uncertainty and noisy data.

The hybrid system developed by Shavlik and Towell used explanation based learning to develop a set of rules from a small training dataset, and used the neural network to refine the rules using additional data for training. Again, the hybrid classifier performed better than either of the individual classifiers used alone.

Battiti and Colla [10] used statistical measures to assign votes to classifications produced by different neural networks. In particular they found that combining the classifications of networks trained with different attributes, different topologies and using different learning algorithms saw the greatest improvements in classification accuracy.

Drucker and Cortes [42] also believe that committees of classifiers perform better than single classifiers. Boosting was used to combine the results of classifiers – each classifier is trained on data that has been filtered by a previously trained classifier. The outcome is to identify, and then remove the misclassified or low signal to noise ratio cases from the training set.

The error rates for character recognition reported by Drucker and Cortes were reduced to 0.7% for neural networks with boosting, from 1.6% for networks without boosting[4]. Similar results have been reported for C4.5 classification across a number of different domains [70].

Real world data typically contains noise, is subjective or can contain contradictory information [70]. He and Huang [70] used neural networks with boosting to classify credit card data – real world data that is noisy, subjective and has contradictory information – and were again able to reduce classification error rates.

For small datasets boosting is difficult as we may end up removing information that could be used in the classification. Pre-processing the data in some way, such as discussed in Chapter 6 is preferable as we are removing some of the noise and contradictory information, without removing the relevant information.

There is ample evidence in the literature to suggest that significant improvements can be obtained by combining the results of more than one classifier (see also [121,

[4]In character recognition domains very high accuracies are already possible and error reduction from 1.6% to 0.7% is significant

27, 53, 79, 83, 15, 19, 7, 73, 3, 41]). The aim for combining classifiers in the context of this work is to make use of the best properties and results of each classifier, and mitigate the effects of inadequate, noisy data.

8.5 Agreement Classification

An approach for combining binary classifications was developed as part of this work, and was first published in [99]. The technique, called agreement classification, only gives a classification for a specific pixel if all binary classifiers are consistent in their classifications. It is based in the idea that a classification system should reject cases that have a high probability of being misclassified [10].

As an example, if a binary neural network classifier gave the following outputs:

Classifier	Binary NN Output	Binary NN Classification
tree	0.92	tree
urban	0.01	not urban
water	0.25	not water

would result in a pixel being classified as `tree` – the classifications of all three binary classifiers support the same conclusion. Whereas a classification with the following outputs:

Classifier	Binary NN Output	Binary NN Classification
tree	0.93	tree
urban	0.87	urban
water	0.31	not water

would result in a pixel being classified as `unknown` – the classifications of the three classifiers do not support each other.

An agreement classification can be used to combine the binary classifications. They are generated by giving a class to a pixel only when all the classifiers agree

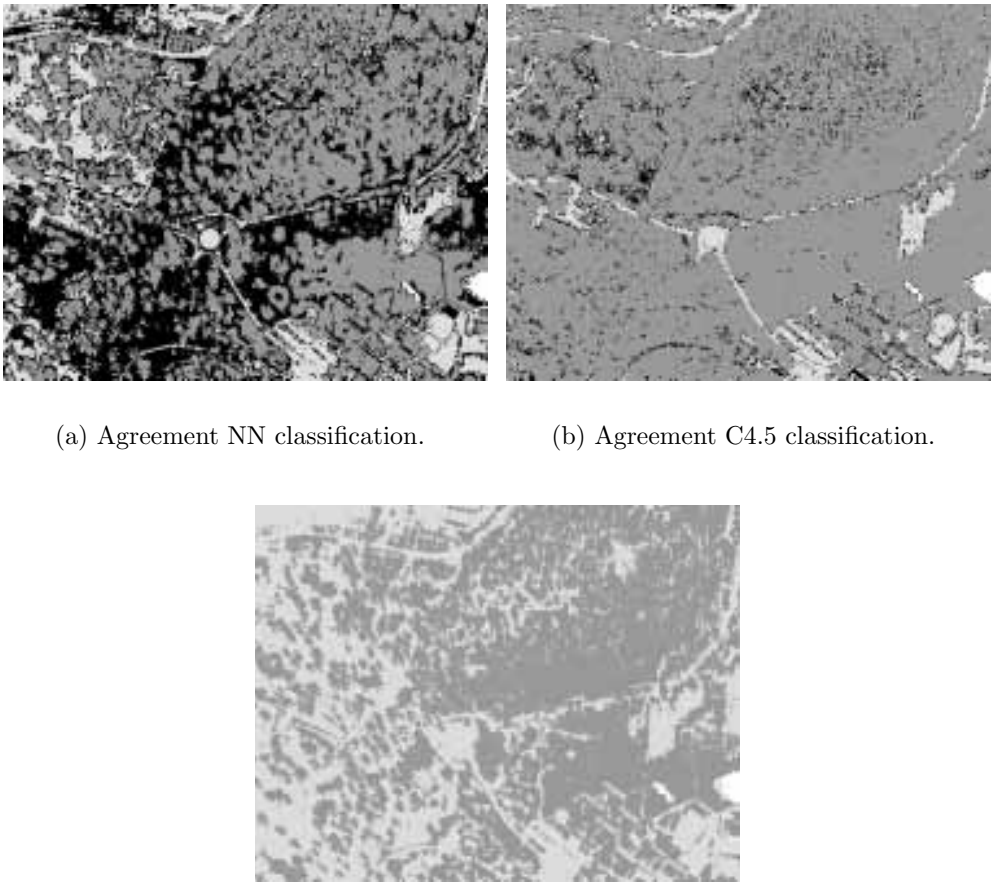
on the class membership of a given case. If one or more of the classifiers gives a different class for a pixel an `unknown` classification is given.

Battiti and Colla state that disagreement between individual classifiers is seen as a symptom of an uncertain classification [10]. Accepting a classification only when all classifiers agree ensures that only pixels with characteristics consistent with the known members of a given class are classified, and reduces the effect of errors in an individual classifier.

A majority vote in an agreement classification is not used as it requires the assumption that individual classifiers are accurate, which may not be the case with small noisy datasets.

Figure 8.8 shows the agreement classifications generated from each of the classifier systems.

Unfortunately, there has been little change in the classification of the entire image. Each of the classification systems gave a reasonably consistent result over the three binary classifications. However, it is clear that each classifier has given very different classified images. We can use this result to improve our overall classification accuracy further.



(a) Agreement NN classification.

(b) Agreement C4.5 classification.

(c) Agreement IBL(k=3) classification.

Figure 8.8: Agreement classifications for binary classifiers.

		Classified As			
		unknown	trees	urban	water
Class	trees		16		
	urban	4		19	
	water				24

Table 8.6: Agreement classification between binary neural network, C4.5 and IBL(k=3) classifications for the test error.

8.6 Multi-Strategy Agreement Classification

Different classifier systems can provide complimentary results which when integrated can support each other as well as compensate for each others weaknesses [106]. By combining classifications from a number of different classification systems we are able to remove some of the misclassified pixels.

Different classifications can be generated in two ways.

Varying the classifier system. A variety of classifiers can be trained on the same dataset. We have done this here by using neural networks, C4.5 and IBL.

Varying the training data. We can vary the training data by grouping different cases together into the known classes. For example, a dataset that contains data for all classes and a dataset that contains only two classes, as we have already done with the multi-class and binary classifications.

Any of the classifications generated using these or other approaches can be combined using agreement classification.

8.6.1 Different Classifier Systems

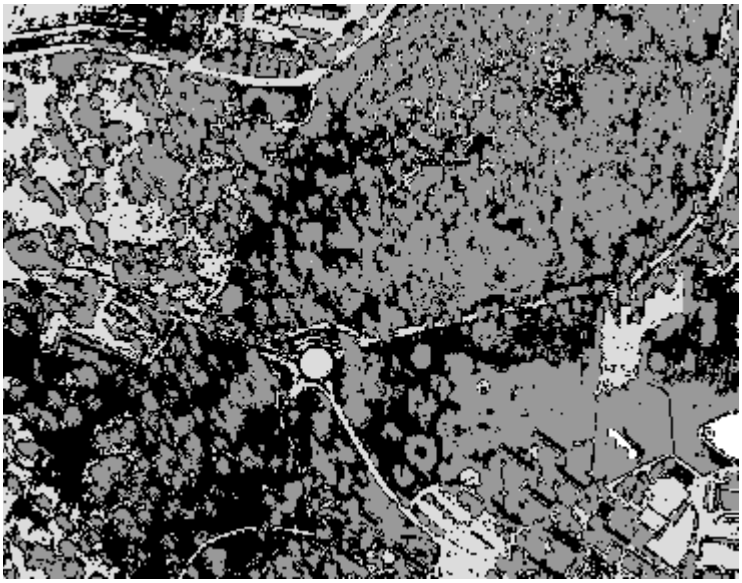
The agreement classifications that were generated from the binary neural network, binary C4.5 and binary IBL classifications (as shown in Figure 8.8) were combined in a further agreement classification. The error on the test set can be seen in Table 8.6.

For the test set we see that none of the pixels were misclassified and this is due to the neural network classification. That is, the C4.5 and IBL binary classifications did not produce classifications inconsistent with the neural network classifications – the majority of the inconsistent classifications were already given a class of `unknown` by the neural network agreement classification. The original test set, that contained `grass` pixels, were given the same classifications as were given in the agreement classification for the binary neural networks.

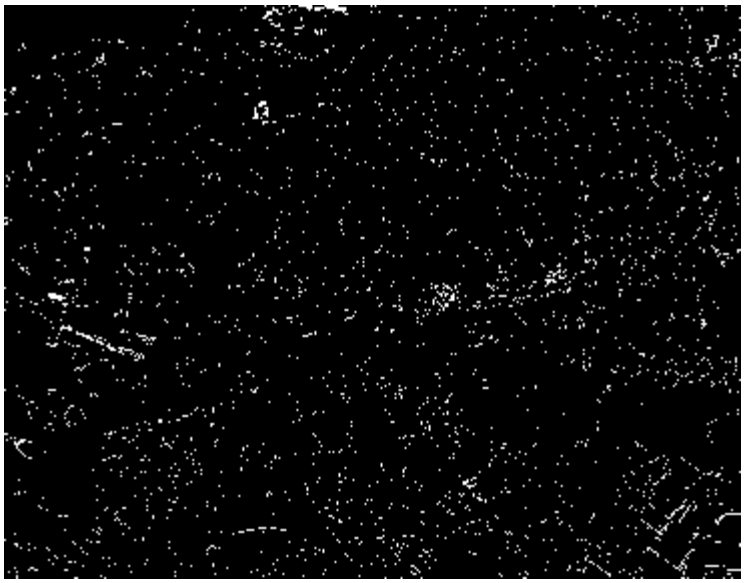
The classification of the entire image is almost identical to the agreement classification of the binary neural networks, and can be seen in Figure 8.9(a). In spite of its similarity to the classification shown in Figure 8.8(a), 4% of the `unknown` pixels were actually due to inconsistent classifications between the three classifiers.

If we look at the 4% of the pixels that were given inconsistent classifications between the three classifiers we find that they lie mostly on the boundaries of objects (see Figure 8.9(b), note that the boundaries between buildings stand out in particular). This is due to rectification errors and mixtures of spectral information at class boundaries. Agreement classification allows us to reduce the effects of these boundary pixels in the training set and will go some of the way to producing a more accurate classifications.

In spite of some misclassifications being removed there are still a large number of pixels with incorrect classifications. For example, the large number of pixels classified as `urban` in the top left hand corner of the image should be classified as `unknown` as they are in the `grass` class. This means that the models that have been generated by the three classifiers are not sufficiently different to improve classification accuracy to a high enough degree.



(a) Agreement classification.



(b) Inconsistently classified pixels.

Figure 8.9: Agreement classification between binary neural network, C4.5 and IBL(k=3) classifications.

		Classified As			
		unknown	trees	urban	water
Class	trees		16		
	urban		1	22	
	water				24

Table 8.7: Test set error for agreement classification between multi-class and binary C4.5 classifications.

8.6.2 Different Training Datasets

Different training datasets can provide a different view of the area to be classified and so produce a different classification. Without generating new data, different training datasets can be quite simply generated by grouping the existing data into different subsets. This has already been demonstrated with the multi-class and binary classification schemes. Each uses the same cases in the training set, however, the cases are grouped into classes in different ways. This forces the classifier to generate a different model of the data, even if the same underlying classification system is being used on the same actual data.

The previously generated C4.5 multi-class (Figure 8.3(a)) and C4.5 binary classifications (Figure 8.8(b)) were combined using agreement classification. The error on the test set was 1.6% as before (see Table 8.7). Of the 24 `grass` pixels from the original test set, 12 were given an `unknown` classification, 11 a `tree` classification and one an `urban` classification. This is a significant improvement over all previous classifications – 50% of the misclassified pixels have been removed.

As can be seen in Figure 8.10, the map produced is also an improvement over the other approaches discussed. Most of the `grass` class has been removed by giving them an `unknown` classification. Some areas in the bottom left hand quadrant remain classified as `trees`, and some as `urban` in the top right. But, overall majority of the `grass` pixels have been removed. All other structures are as clear as for the classifications trained on all four classes (as seen in Figure 8.1).

The use of differently trained classifiers causes different models of the data to be found. The characteristics that describe prototypical cases of each class are refined when such classifications are combined using agreement.

Figure 8.10: Agreement classification between the multi-class and binary C4.5 classifications.

8.7 Comparison with Cross-Validation

A commonly used technique used to determine classification accuracy is n-fold cross validation [85]. The available training data is split into n mutually exclusive subsets of approximately equal size. The classifier is trained n times on all of the data in $n-1$ of the partitions and tested on the remaining partition. The error estimate is the average error over each of the n disjoint test partitions. If the accuracy estimate is highly variable over each of the test partitions the error estimate provided by sub-sampling techniques are likely to be unreliable [85].

Early experiments using cross validation on small, noisy datasets confirmed the high variability in error estimates across different classifiers trained on different sub-samples of the data. However, using cross validation in combination with the techniques as discussed here do not add any additional value as the error estimates are more consistent when using agreement classification for a given classification task. In addition the error estimates using cross validation were found to be comparable to that obtained by the classification framework described in this thesis, and requires an additional $n-1$ classifiers to be trained. This increases the training times substantially.

For these reasons cross validation was not used in this work.

8.8 Discussion and Conclusions

In this chapter we discussed techniques for reducing the numbers of misclassifications – firstly, by thresholding neural network output values, and secondly by generating a number of classifications that are combined by agreement classification. In particular, binary classifiers have allowed two opportunities for reducing the number of misclassification – first, by training classifiers on a simple task, and then when the results are combined using an agreement classification. The performance of each classifier can be further improved by using attribute selection, as discussed in Chapter 6.

The reasons for inconsistent classifications being given to a pixel can include:

- Inconsistently classified pixels may belong to an existing class but have sufficiently different characteristics to other members of that class. Differences, for vegetation mapping for example, may be due to factors such as different light or nutrient levels, or noise and errors in the training dataset.
- An area may not be classifiable because it belongs to a class not represented in the training dataset.

Partial classification of an image, to remove the misclassified pixels, is not seen as a disadvantage, rather as an advantage. Areas that are classified have a better chance of being correctly classified, while those that have not been classified are simply those needing further investigation.

Combining the results from different classifiers may seem to increase the error rate on the test set in the sense that some pixels are given an `unknown` classification. However, these pixels have not been misclassified and so are not considered in error. If we look at the classification over an entire image we see a significant reduction in the number of misclassifications. This approach is seen as an improvement over simpler classification approaches as we have an increased level of confidence that the error rate on the test set can be translated directly to an error rate over the entire image.

Most importantly any classification system can be included in an agreement classification as it is a general technique that uses previously classified data only.

An important consequence of the approach described in this chapter is that, rather

than classifying an entire image with limited accuracy (varying with class and conditions) we identify areas we know we can classify relatively accurately, and say nothing about the rest of the image. This also means that we no longer have to assume we have complete information. That is, we can assume we do not have sufficient labelled training data to entirely describe the domain.

When only a small number of training cases are misclassified, due to there being a large amount of high quality training data, the approach suggested in this thesis is certainly overkill. Rather, these techniques are specifically for domains with small and noisy datasets, where the number of misclassifications is going to be high.

A simple dataset was used in this chapter, firstly, so that the improvements demonstrated could be attributed to the technique, not vagaries of the data. Secondly, a small number of training cases were deliberately chosen to simulate the real world situation where only small amounts of training data are available.

The techniques described here can also be used to identify some types of data errors. In the situation where a training case is given an incorrect classification by a classifier it should be investigated further. This kind of error may indicate an error in the data or the need to investigate why such cases differ from the general properties of the given class. In this situation the classification techniques can provide a data mining function, and be used to analyse the available data.

A further problem with the vegetation mapping domain in particular is that it is generally not possible to quantify the error over the entire image accurately. The error rate on a test set does not always translate directly to a reasonable estimate of the error rate over the entire image, due to the inadequacy of the training data. So far we have been evaluating the error on the entire image using qualitative assessment, which is obviously a source of errors. In the next chapter we look at simulating remotely sensed data to allow us to better quantify classification error and so further investigate the application of attribute generation, attribute selection and agreement classification.

Chapter 9

Simulating Remotely Sensed Data

A major concern for classification of remotely sensed data is quantification of the accuracy of the training data and that of classifications generated from it. The training data itself may contain errors or noise. The classifier generated from this data often also produces misclassifications. From the training data, typically containing only a few hundred cases at best, we are trying to classify an entire image consisting of hundreds of thousands of pixels. How can we quantify the true error?

In the absence of a training dataset with a large number of cases, a low error rate on a test set does not necessarily mean a low error rate over the entire image. This is of particular importance when we wish to compare the performance of different classification techniques. This was clearly demonstrated in the previous chapter – all test sets had very low error rates and it was only when the classifiers were applied to an entire image that it became clear that all varied quite substantially in their classifications and which one was the superior classification.

As there were no datasets with the required properties that were publicly available at the time of writing, simulating a remotely sensed image was seen as the only viable alternative. A simulated image is one where the class of all pixels are known and so can help us to quantitatively compare the classifications generated by different classification systems. An image can be simulated by using the statistical properties of training data from an actual image.

Chen et al [22] simulated remotely sensed images to test the performance of a dynamic learning neural network. An unsupervised classification of the image was generated using ISOCLASS clustering [9] for three bands of a 512x512 frame from a SPOT image. This generated an 8 class classification which was used as the basis for a simulated image. The mean values for each class in each band were calculated and the pixels in the simulated image were assigned random numbers having the mean of the given class and a variance of four.

In the classifications carried out by Chen et al [22] error rates of between 1% and 10% were reported for the simulated remotely sensed data. These results carried through to the classification of a real image with an estimated error rate of 8%.

By using a variance of four the generated spectral values for each of the classes have been tightly constrained, possibly helping to achieve the very low error rates. The larger the variance, the larger the overlap between classes is going to be. That is, the classes will be more easily distinguishable for a small variance, especially when there are only a small number of broad classes, such as water and vegetation. Images produced in this way do not reflect the variance seen in a real image very well.

A real image will also show correlations between pixels. The spectral value for a given pixel in a given band will be correlated to the spectral values of its neighbouring pixels. This is because most pixels will contain mixtures of objects and so contain a mixed spectral response and most scenes show gradual changes between pixels.

In addition, the spectral values for each pixel in an image will be correlated across each of the bands. The covariance's between spectral bands in the approach used by Chen et al [22] are taken care of in that the standard deviations for the simulated image are very small. This means that the simulated values will show similar covariance's to the original data.

In this chapter we look at extending the method for simulating a remotely sensed image used in [22] to include these additional statistical characteristics.

9.1 Method for Generating Images

The simulated image generated here was based on the *CSU* image (see Figure 9.1).

The process for simulating a remotely sensed image was as follows.

1. **Generate training data.** A set of classes to be used for the simulated image are identified in a real image. Pixels that can be reasonably accurately classified are extracted from the existing image, giving a set of classified cases. The mean spectral value for each class over each band and the covariance's between bands for each class are then calculated.

   Fifteen high level classes were identified in the *CSU* image. These classes were chosen because consistent areas of pixels could be easily identified within the image. At least 100 pixels were extracted for each of the classes and the means and covariance's calculated. The classes and statistical characteristics can be seen in Table 9.1.

2. **Generate a classified image.** An image of size $r \times c$ is generated where the class of each of the pixels is known. This can be easily done by mapping the desired classes to the classes given for an unsupervised classification of a real image.

   The classified image used here can be seen in Figure 9.4 and was generated from an AutoClass classification of the *CSU* spectral data.

3. **Generate standard normal random numbers.** For each pixel in each band in the simulated image a random number with mean 0 and standard deviation 1 is generated. The number of bands in the simulated image will be the same as the original image the training data was derived from.

   The *CSU* image contains four spectral bands and so the simulated image will also have four bands. For each of the pixels in the classified image, four random numbers were generated that will become the four spectral values. The Box-Muller method was used here to generate random normal numbers [84].

4. **Band Covariance.** The spectral response for a given pixel will be correlated across all bands for a given class. Using the algorithm described in Figure 9.3, we can generate dependant numbers with given distributions from random numbers.

   The four independant random numbers generated for each pixel in the classified image are correlated using the algorithm in Figure 9.3 using the means

and covariance's of the class for that pixel. The values for each pixel resulting from this transformation will be correlated between the bands and are in the range 0 to 255.

At this stage we have a simple simulated remotely sensed image with 15 classes and four spectral bands.

5. **Pixel spatial correlation.** The spectral values between neighbouring pixels within a given spectral band in an image are also correlated. To achieve this in the simulated image the spectral value of a pixel in a given band is replaced by the weighted sum of its spectral value and that of its neighbouring pixels.

   For the simulated *CSU* image the weighted sum was as follows. Each pixels value is replaced by $0.8r + 0.2m$, where r is the pixels current value and m is the average spectral value of the 8 neighbouring pixels. This also serves to blur the boundaries between classes by averaging the spectral values.

6. **The addition of noise.** A real image will contain at least some noise due to factors such as atmospheric effects and properties of the sensor being used. The simulated image generated so far is statistically well behaved and so a small amount of noise should be added. This can be done in a number of ways.

   One source of noise in an ABVS image is the the video camera lenses. This noise radiates out in a circular pattern from the centre of the lens. Pixels at the edge of the image cover a larger area on the ground than the pixels at the centre of the image and so are slightly darker. The circular noise pattern produced by a camera lens was simplified to a darkening across the image for this work.

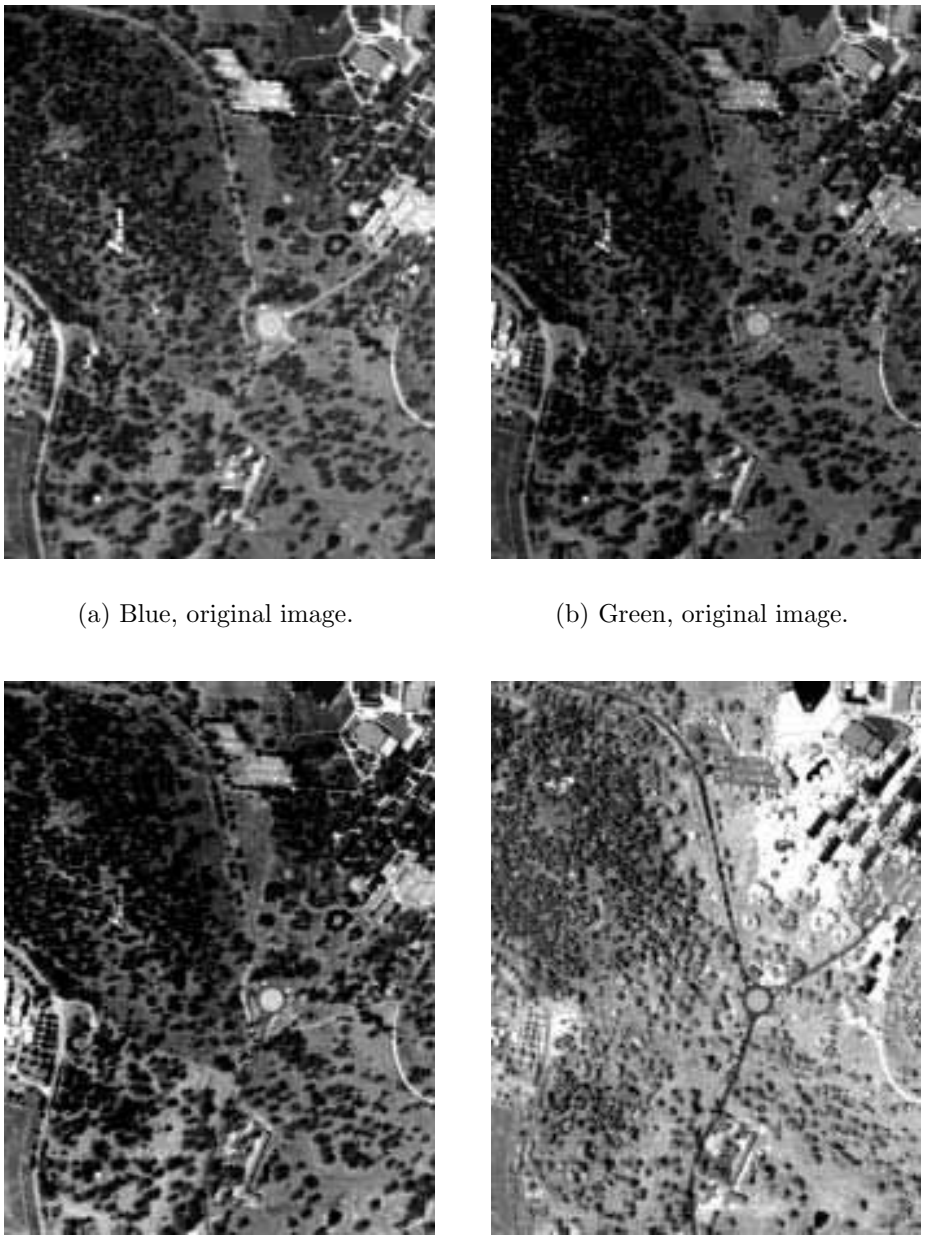
   A graduated image, as seen in Figure 9.2, was used to scale the spectral values in the simulated image and make the pixels darker at one end of the image. The graduated image was the same size as the simulated image with values in the range 192 to 255.

   Each pixel in the simulated image was replaced by the value

   $$\frac{r_{ij}m_{ij}}{255}$$

   where r_{ij} is the spectral value of the pixel in row i and column j and m_{ij} is the value in the corresponding pixel in the graduated image.

The image generated using this method can be seen in Figure 9.5.



(a) Blue, original image.

(b) Green, original image.

(c) Red, original image.

(d) NIR, original image.

Figure 9.1: Original remotely sensed image.

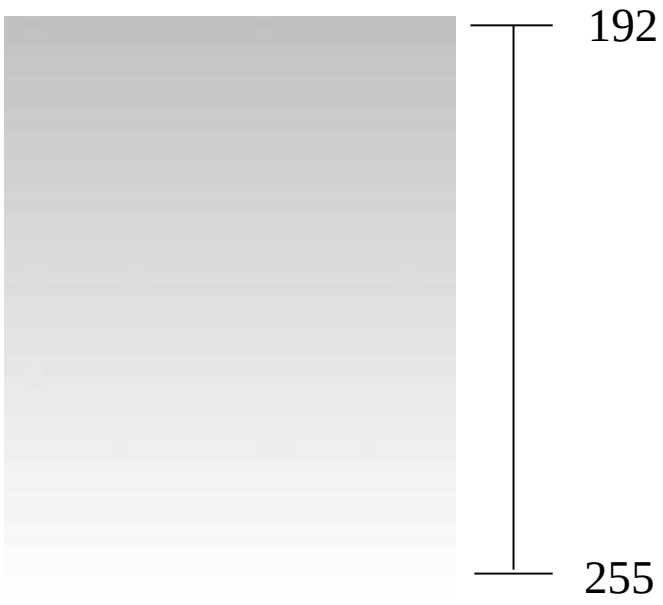


Figure 9.2: Noise levels added to the simulated remotely sensed data – the value in the graduated image was used to reduce the magnitude of the values in the simulated image.

Generating dependant random numbers.

Given n independant normal random variables X_i (mean 0 and standard deviation 1) it is possible to generate n dependant normally distributed variables Y_i (means μ_i and covariance c_{ij} between variables Y_i and Y_j) as follows.

$$\mathbf{Y} = \mathbf{AX} + \mathbf{M}$$
$$\mathbf{AA}^T = \mathbf{C}$$

where $\mathbf{A}$ is a triangular matrix, $\mathbf{M}$ is the vector of means and $\mathbf{C}$ is the covariance matrix.
If we wish to generate four bands of data four dependant numbers (y_i) can be generated from four independant random numbers (x_i) as follows.

$$y_1 = \mu_1 + a_{11}x_1$$
$$y_2 = \mu_2 + a_{21}x_1 + a_{22}x_2$$
$$y_3 = \mu_3 + a_{31}x_1 + a_{32}x_2 + a_{33}x_3$$
$$y_4 = \mu_4 + a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4$$

The following simultaneous equations need to be solved for the given covariance matrix to get the coefficients a_{ij}.

$$a_{11} = \sqrt{c_{11}} \qquad a_{21} = \frac{c_{12}}{a_{11}}$$
$$a_{31} = \frac{c_{13}}{a_{11}} \qquad a_{41} = \frac{c_{14}}{a_{11}}$$
$$a_{22} = \sqrt{c_{22} - a_{21}^2} \qquad a_{32} = \frac{c_{23} - a_{21}a_{31}}{a_{22}}$$
$$a_{42} = \frac{c_{24} - a_{21}a_{41}}{a_{22}} \qquad a_{33} = \sqrt{c_{33} - a_{31}^2 - a_{32}^2}$$
$$a_{43} = \frac{c_{34} - a_{31}a_{41} - a_{32}a_{42}}{a_{33}} \qquad a_{44} = \sqrt{c_{44} - a_{41}^2 - a_{42}^2 - a_{43}^2}$$

Further details can be found in [84].

Figure 9.3: Generating correlated random numbers.

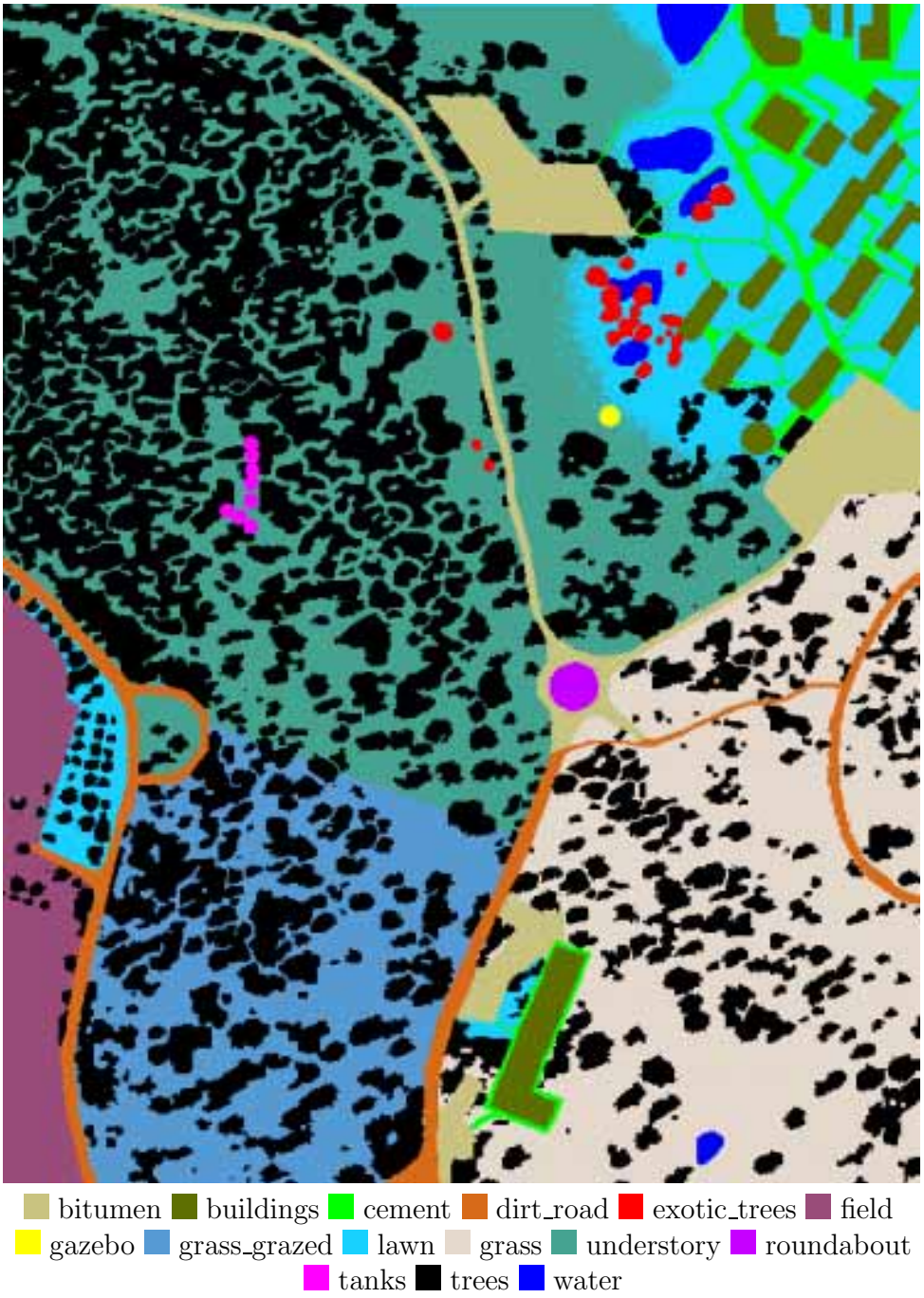
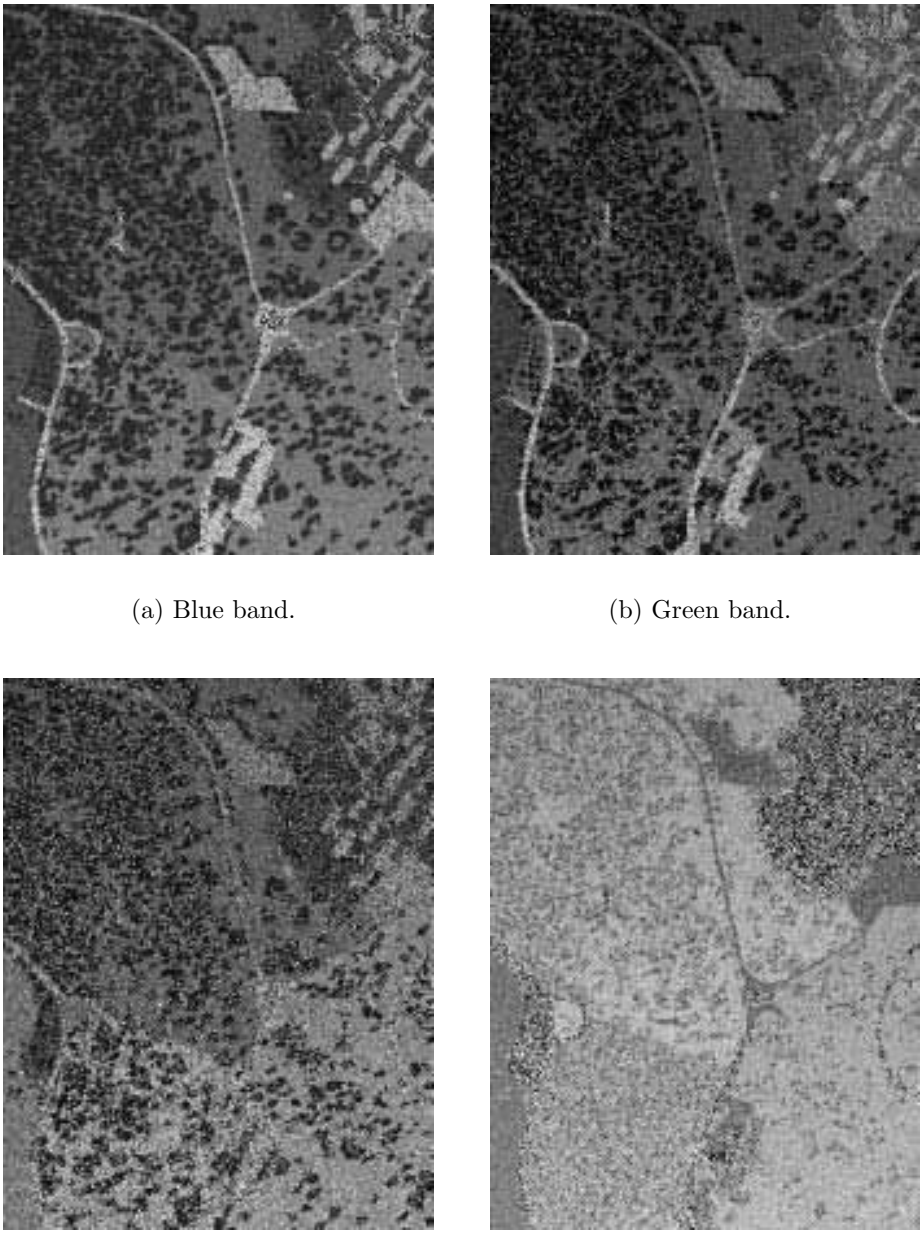


Figure 9.4: Classification of the simulated image.

Class	Mean				Standard Deviation			
	blue	green	red	nir	blue	green	red	nir
bitumen	177	112	128	107	38	36	41	31
buildings	171	158	178	135	43	33	47	30
cement	248	220	230	215	23	53	55	35
dirt.road	192	157	198	224	51	52	48	33
exotics	31	54	9	223	21	30	15	55
field	78	57	104	122	17	16	26	19
gazebo	185	155	146	134	40	25	25	33
grazed	116	88	138	150	28	29	40	29
lawn	67	89	41	250	25	22	30	22
grass	120	89	130	173	18	16	24	25
understory	104	82	93	192	24	18	27	27
trees	40	13	12	147	22	14	15	65
water	57	51	29	16	39	35	32	49

Table 9.1: Statistical characteristics of the classes used in image generation.



(a) Blue band.

(b) Green band.

(c) Red band.

(d) NIR band.

Figure 9.5: Simulated remotely sensed data.

9.2 Comparison of the Simulated Image and the Original Image

The characteristics of the two images are compared here.

9.2.1 Statistical Characteristics

The statistical characteristics[1] for each band can be seen in Tables 9.2 - 9.5. Each table shows the minimum, maximum, mean (μ), median values and the standard deviation (σ) for each class in the image.

As expected both images show similar statistical properties, although the simulated image values are generally slightly lower. This is due to the addition of noise, reducing the values in the simulated image.

[1]The statistical characteristics for the original image use only the class data that was extracted, not the entire image for obvious reasons.

Blue										
	Original Image					Simulated Image				
Class	Min	Max	μ	Median	σ	Min	Max	μ	Median	σ
bitumen	12	255	177	176	38	1	229	146	151	44
buildings	27	255	171	161	43	1	250	153	153	38
cement	12	255	248	255	23	1	199	96	50	85
dirt.road	72	255	192	198	51	1	238	146	170	72
exotics	1	124	31	27	21	7	83	33	32	15
field	27	154	78	79	17	58	140	84	83	15
gazebo	91	243	185	191	40	11	200	154	154	29
grazed	1	255	116	124	28	81	214	123	121	24
lawn	1	213	67	61	25	31	140	63	62	18
grass	16	191	120	124	18	92	185	123	122	15
understory	5	202	104	102	24	64	159	96	95	17
trees	1	131	40	39	22	13	103	42	41	17
water	20	236	57	35	39	13	158	56	53	26

Table 9.2: Statistical characteristics of the classes in the simulated image for the blue band.

Green										
	Original Image					Simulated Image				
Class	Min	Max	μ	Median	σ	Min	Max	μ	Median	σ
bitumen	1	255	112	120	36	3	224	108	105	33
buildings	9	255	158	154	33	1	248	140	141	47
cement	16	255	220	247	53	1	200	102	119	62
dirt.road	1	255	157	154	52	1	237	146	150	57
exotics	1	139	54	57	30	12	159	57	55	27
field	5	124	57	53	16	32	144	66	64	20
gazebo	120	202	155	154	25	110	186	139	136	20
grazed	1	255	88	91	29	28	246	102	100	34
lawn	1	191	89	91	22	44	162	83	81	22
grass	1	154	89	91	16	60	176	95	94	19
understory	1	154	82	83	18	45	169	79	77	18
trees	1	79	13	12	14	1	226	30	20	48
water	1	191	51	31	35	2	163	55	51	33

Table 9.3: Statistical characteristics of the classes in the simulated image for the green band.

Red										
	Original Image					Simulated Image				
Class	Min	Max	μ	Median	σ	Min	Max	μ	Median	σ
bitumen	1	255	128	135	41	1	230	114	113	52
buildings	1	255	178	165	47	1	251	119	130	66
cement	5	255	230	255	55	1	200	103	110	59
dirt.road	5	255	198	202	48	1	242	127	146	71
exotics	1	98	9	12	15	1	215	59	29	74
field	1	195	104	102	26	12	242	120	117	36
gazebo	79	180	146	150	25	88	191	130	126	28
grazed	1	255	138	146	40	1	255	145	146	57
lawn	1	198	41	39	30	1	216	61	54	45
grass	5	254	130	131	24	1	253	143	140	36
understory	1	165	93	94	27	8	209	96	93	33
trees	1	91	12	16	15	1	226	53	31	68
water	1	221	29	12	32	1	203	61	49	52

Table 9.4: Statistical characteristics of the classes in the simulated image for the red band.

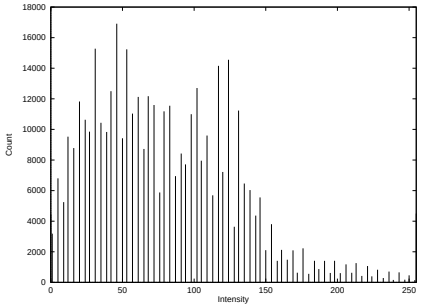
NIR										
	Original Image					Simulated Image				
Class	Min	Max	μ	Median	σ	Min	Max	μ	Median	σ
bitumen	1	202	107	117	31	2	213	101	98	30
buildings	1	255	135	139	30	1	251	124	124	38
cement	12	255	215	221	35	1	196	107	133	65
dirt.road	24	255	224	228	33	1	243	129	174	84
exotics	5	255	223	255	55	1	217	108	130	67
field	31	255	122	120	19	95	198	127	126	18
gazebo	46	172	134	143	33	27	210	126	114	43
grazed	5	255	150	154	29	1	254	158	159	46
lawn	20	255	250	255	22	1	216	95	38	88
grass	1	255	173	176	25	5	251	176	174	19
understory	16	255	192	191	27	1	222	170	169	23
trees	1	255	147	146	65	1	225	137	139	48
water	1	255	16	76	49	1	203	89	80	61

Table 9.5: Statistical characteristics of the classes in the simulated image for the near infra-red band.

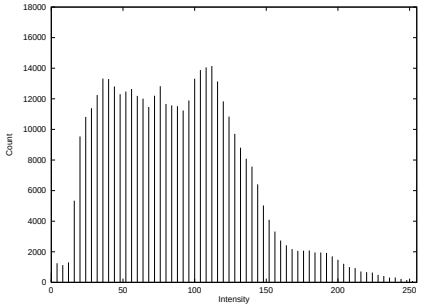
9.2.2 Histograms

Histograms for each band of each image can be seen in Figure 9.6. The histograms, using the data from the entire image, show that the spread of spectral values are similar.

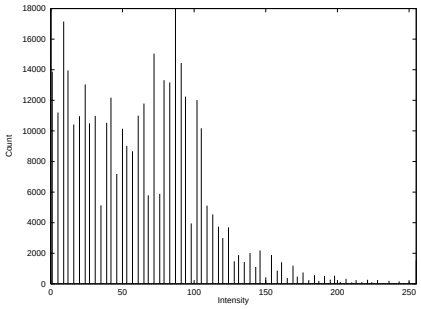
Note that the values for the simulated image are spread over all possible 256 spectral values. However, the original image does not have all spectral values represented in the image due to the method of data acquisition. This means that the histograms for the simulated image would have smaller counts spread over all values. To reduce the spread for the simulated image every three values have been grouped into the same bin in the histogram.



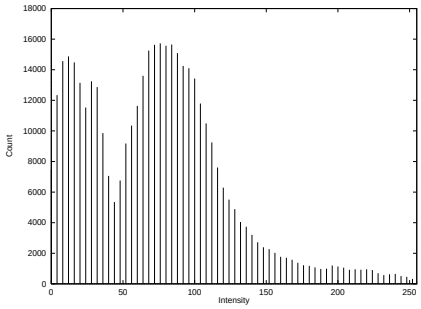
(a) Blue, original image.



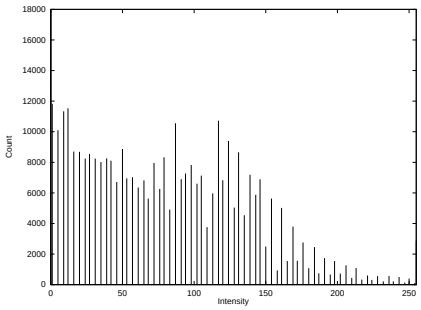
(b) Blue, simulated image.



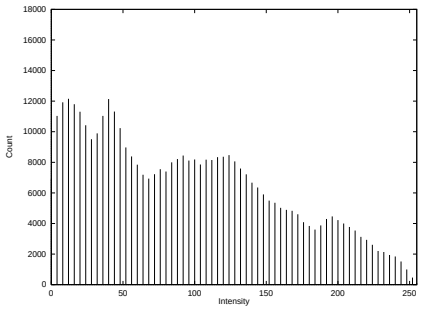
(c) Green, original image.



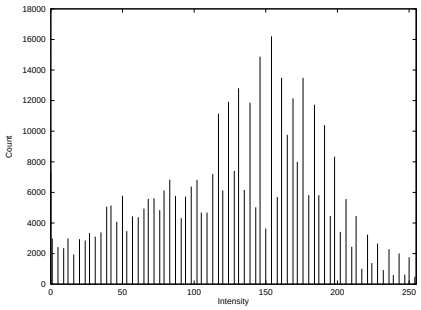
(d) Green, simulated image.



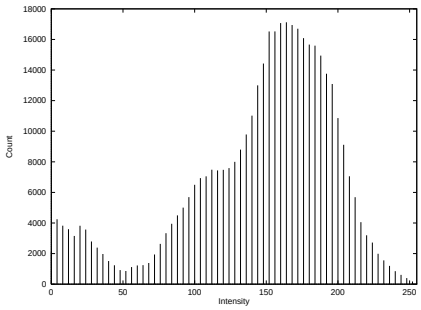
(e) Red, original image.



(f) Red, simulated image.



(g) NIR, original image.



(h) NIR, simulated image.

Figure 9.6: Histograms of the original spectral data and simulated data.

Class	Class id.	# Pixels in Image	% Pixels in Image	# Training Cases	# Stop Cases	# Test Cases
bitumen	1	19552	4.6%	44	21	22
buildings	2	12115	2.7%	38	18	19
cement	3	8427	2.0%	34	17	17
dirt.road	4	9501	2.2%	28	14	14
exotics	5	1873	0.4%	31	15	16
field	6	13419	3.2%	25	12	12
gazebo	7	137		19	9	9
grazed	8	33240	7.9%	42	20	21
lawn	9	30869	7.3%	34	17	17
grass	10	61806	20.9%	42	21	21
understory	11	88348	14.6%	44	21	22
roundabout	12	735	0.2%	27	13	13
tanks	13	543	0.2%	18	9	9
trees	14	138957	32.8%	37	18	19
water	15	3516	0.8%	33	16	17

Table 9.6: Number of cases in each dataset.

9.2.3 Classification Performance

Finally, we do a simple classification of the simulated image to test its behaviour when classified.

The classification carried out here is a simple four attribute multi-class classification. The attributes are the four spectral bands from the simulated image that can be classified into one of the 15 classes as shown in Table 9.6. A total of 985 cases were randomly selected, as described in Section 3.3, from the image and split into three sets.

The C4.5 and IBL classifications gave the kind of results that would be expected (see Table 9.7). These test set error rates are not unusual for this domain, but are still high. Table 9.8 shows the number of pixels given each classification from the test set.

Classifier	Test Error	Image Error
C4.5	44.0%	56.6%
IBL(k=1)	32.6%	73.6%
IBL(k=3)	44.7%	70.2%

Table 9.7: Classification error on the simulated image data.

		Classified As														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Class	1	16	1	1				1	1			1				1
	2	5	7		1	1			2		1	1				1
	3	4	2	5	2	2							1	1		
	4	1			9				1				3			
	5			1		11				1					3	
	6						7		1			2			1	1
	7		2					4	2		1					
	8	1	2				1		10	1	3	2			1	
	9			2		4				7						4
	10								4	2	14	1				
	11						5					15			1	1
	12	1			2	1							8			1
	13	1		3										5		
	14			1		1									16	1
	15	2		1		3				2		2	1		1	5

(a) C4.5 Classification.

		Classified As														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Class	1	15	3	1				1	1		1					
	2	3	5	2				1	4				1	2		1
	3		1	9	2	1							2		1	1
	4			3	9						1			1		
	5					11				2					3	
	6						8		1			2			1	
	7	1						6	2							
	8		1		1		2		14	1		2				
	9									12					1	4
	10								2		19					
	11						1					21				
	12			3	1								8		1	
	13		1	2										5		1
	14					1									17	1
	15	1		2			2			3		1				8

(b) IBL(k=1) Classification.

		Classified As														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Class	1	15	5					1			1					
	2	5	5	4				1	2				1			1
	3		1	7	1	3		1				1	2			1
	4			4	7				2		1					
	5					9				2			1		3	1
	6						7		1			3			1	
	7	1	3					3	2							
	8	1			1		1		12		2	3				1
	9					1				9					1	6
	10								1		17	3				
	11						3		1	1		17				
	12			2	1	1							7	2		
	13		1	3									1	3		1
	14									2					14	3
	15			1			2			6		2			1	5

(c) IBL(k=3) Classification.

Table 9.8: The number of pixels given in each class for the test set.

Over the entire image we see a large increase in the error rate as compared with the test set error, as seen in Table 9.9. The number of incorrectly classified pixels has increased, but with no particular pattern of misclassification.

The neural network that was trained on the simulated data had three layers, each with four nodes. The four input attributes, which were the four spectral values, were scaled to be between 0 and 1. The 15 target classes were mapped to a four digit binary number, giving the four output values for the four output nodes, as seen in Table 9.10.

The neural network classification performed very poorly and almost all cases were classified as `cement` (class id 5), giving an error rate of 93% on the test set. However, this is to be expected in a neural network classification as there were only 4 inputs and 15 classes to be identified.

This neural network was unable to distinguish any of the class characteristics in the input data. All but a few of the output values were identical to within 2 significant digits. While the other classifiers were able to distinguish at least limited features in the input data, the neural network could distinguish none. Thresholding output values is of no use in this case due to the lack of separation in output values.

Even if we consider the C4.5 and IBL test set error as acceptable, we see that the error over the entire image is considerably higher. The problem with this type of approach is that each classifier is trying to distinguish a large number of classes with a small amount of information.

Such poor classification results indicate that the classification task is too complex and not enough information of sufficient quality is available to the classification system. And yet these types of classifications are not all that unusual within the remote sensing and vegetation mapping domains.

		Classified As														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Class	1	7145	2944	2114	841	80	310	37	2398	42	1552	1642	125	91	4083	261
	2	4947	2528	1510	989	29	40	29	2216	27	490	243	41	61	8	13
	3	298	302	1312	1035	147	0	4	15	0	0	0	85	140	1019	54
	4	1131	1419	604	2520	0	34	6	4660	13	1384	125	33	25	1	19
	5	93	16	386	217	435	1	0	0	7279	0	39	35	18	25707	684
	6	59	46	0	0	0	10004	0	2372	0	1352	8278	0	0	548	88
	7	2945	1748	494	319	0	127	33	2141	2	1199	110	21	25	0	0
	8	736	1272	0	593	5	905	18	11593	777	32879	22901	0	0	1560	57
	9	12	4	31	97	56	458	0	590	6459	225	1917	3	4	6659	259
	10	234	516	0	70	0	35	8	3047	27	18391	8604	0	0	0	7
	11	223	532	0	219	3	587	0	2379	2850	4190	40833	0	0	1658	182
	12	1179	510	1079	1930	24	0	1	4	578	0	9	290	108	3766	66
	13	139	28	99	165	46	0	1	1	404	0	0	32	16	514	29
	14	111	101	523	222	846	424	0	1649	8560	142	2650	40	31	81656	1173
	15	300	149	275	284	202	494	0	175	3851	2	997	30	24	11778	624

(a) C4.5 Classification.

		Classified As														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Class	1	6712	3276	1752	1399	3	266	57	1957	302	662	2065	91	90	1579	13
	2	1337	1511	694	1356	0	0	15	737	0	39	7	86	56	236	0
	3	402	220	929	964	211	611	1	107	839	0	110	124	88	6034	195
	4	238	257	406	1022	0	0	0	21	0	0	0	56	45	1065	4
	5	369	88	791	234	823	819	0	1106	19707	1269	11788	34	27	19416	1805
	6	2337	1649	73	247	6	1943	16	12517	275	40139	17957	3	3	44	38
	7	464	592	375	298	1	4	3	245	13	0	0	23	15	140	3
	8	823	931	617	1058	0	0	5	1133	3	254	10	41	41	3	1
	9	1153	587	252	250	0	40	0	1159	1100	528	1401	11	14	1	70
	10	207	389	106	609	0	0	5	1452	4	246	2	4	6	0	3
	11	619	642	112	499	1	10	14	4860	102	8086	1472	2	1	13	9
	12	577	130	751	734	33	781	0	222	1476	3	316	117	64	5133	131
	13	144	29	499	285	144	428	0	58	834	0	498	75	37	10590	134
	14	248	129	517	65	598	2068	0	2517	5264	6257	30308	43	29	94510	920
	15	3922	1685	553	481	53	6449	21	5149	950	4323	22414	25	27	193	190

(b) IBL(k=1) Classification.

		Classified As														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Class	1	7847	3610	1761	1466	3	297	71	2424	289	622	1274	96	90	1195	12
	2	1218	1697	849	1616	0	0	14	592	0	48	5	90	69	170	0
	3	375	247	936	981	194	444	2	59	715	0	74	119	78	3054	169
	4	175	197	257	895	0	0	0	17	0	0	0	54	30	910	3
	5	387	77	813	223	834	977	0	1033	19726	1555	11284	38	28	9912	1738
	6	2241	1508	103	244	12	1807	11	11499	293	35692	13507	3	6	25	36
	7	481	557	382	319	0	2	1	209	4	0	0	31	26	101	3
	8	737	912	467	914	0	0	10	1074	3	318	9	36	24	5	1
	9	1030	539	273	265	3	30	0	1130	2448	406	1982	9	14	3	121
	10	271	494	124	715	0	0	8	2135	9	449	8	3	8	0	3
	11	544	648	166	524	0	2	7	5332	141	9402	2802	2	2	20	9
	12	378	99	759	681	40	447	0	187	587	4	464	118	70	7515	84
	13	140	19	479	249	143	507	0	59	866	0	491	65	39	10961	144
	14	251	123	496	61	560	2334	0	2640	4929	6170	31643	42	30	104985	989
	15	3477	1388	562	348	84	6572	13	4850	859	7140	24805	29	29	101	204

(c) IBL(k=3) Classification.

Table 9.9: The number of pixels given in each class over the entire image.

Class	Target NN Output Values
bitumen	0.1 0.1 0.1 0.9
buildings	0.1 0.1 0.9 0.1
cement	0.1 0.1 0.9 0.9
dirt_road	0.1 0.9 0.1 0.1
exotic_trees	0.1 0.9 0.1 0.9
field	0.1 0.9 0.9 0.1
gazebo	0.1 0.9 0.9 0.9
grazed_grass	0.9 0.1 0.1 0.1
lawn	0.9 0.1 0.1 0.9
grass	0.9 0.1 0.9 0.1
understory	0.9 0.1 0.9 0.9
roundabout	0.9 0.9 0.1 0.1
tanks	0.9 0.9 0.1 0.9
trees	0.9 0.9 0.9 0.1
water	0.9 0.9 0.9 0.9

Table 9.10: Neural network class labels.

9.3 Discussion and Conclusions

In this chapter we have introduced a method of generating a simulated remotely sensed image that is realistic enough for us to use in further investigations of the classification techniques so far discussed. In the following chapters we will look at the application of the attribute generation and selection techniques and generating agreement classifications using the simulated image.

The assumption that the data used to generate the simulated image is reasonably accurately classified has been made and clearly there is no way of quantifying how accurate it is. Statistically the simulated image is similar to the original image and it demonstrates enough of the characteristics of a real image to be useful.

Error rates of around 40% on the classification of the test set are reasonably high when compared to that seen on other real datasets, and are on the border of usable results. But it must be kept in mind that we are not necessarily trying to get a high accuracy classification from this data. We are more interested in quantifying the results given by the techniques as discussed in previous chapters.

The classes in the simulated image are reasonably high level classes, but, not a lot would be gained from a more detailed simulation. More detail would mean more consistent classes (i.e. smaller variance), but again, generating accurate statistical data is a problem.

As with a real image there is overlap between the classes in the simulated image as demonstrated by the error rates in classification, and in the statistical characteristics of the image. In a real image this may be due to a number of factors. There may be actual overlap in the spectral characteristics of different classes or there may also be spectral mixing[2].

It must be emphasised that no claim is being made to having produced a genuine remotely sensed image. We have merely tried to mimic some of the behaviour of remotely sensed data to enable us to evaluate the classification techniques described in this thesis. It would not be possible, nor even desirable, to emulate the nuances and detail of a true remotely sensed image. The image generated is, however, a close enough approximation to enable us to make quantitative

[2]That is, it is unlikely that a pixel will contain only one object from a well defined class. This means that the spectral characteristics of a pixel will be a mixture of its components.

comparisons between different classifications and to evaluate the effectiveness of the classification techniques described in this thesis.

The simulated data can be used to demonstrate the techniques introduced in Chapters 4 to 8 to increase the classification accuracy, and to reduce the number of misclassifications over an entire image. This will be discussed in the following chapters.

Chapter 10

Automated Classification and Evaluation

In this thesis we have introduced a variety of techniques that can be used for automatically generating classifications. Firstly, we discussed extracting information from remotely sensed data to highlight information. The information that we extract can be used as additional attributes in a classification. As we can generate a large number of attributes, and existing attributes may contain noise or errors, we then looked at attribute selection to choose the most relevant attributes for a given classification task. Finally, we looked as ways of increasing classification accuracy in remotely sensed images. However, we could only qualitatively assess the accuracy of the classified images, so we looked at simulating remotely sensed data so that we can evaluate the accuracy of a classifier over an entire image.

In this chapter we look at combining all of the techniques introduced in this work to automatically classify the simulated remotely sensed image and assess the accuracy of those classifications. For a given classification task the automated process involves the following steps to generate a classified image.

1. A large number of additional attributes are generated from the remotely sensed data using any number of pre-processing and data mining techniques.

2. A multi-class classifier is trained on all available data and attribute selection is carried out using the wrapper method for C4.5 and IBL classifications and contribution analysis for neural networks.

3. Binary neural network, C4.5 and IBL classifiers are generated for each of the classes in the classification, also using attribute selection. The cases in the training dataset are grouped into two classes, the first containing a single class to be recognised and the second to contain the cases for all remaining classes. The resulting classifications from the individual binary classifiers are combined into a single classification using agreement classification.

4. Final classifications of the image are generated by combining up to four multi-class or binary agreement classifications in a further agreement classification.

A large number of classifications can be generated automatically in this way, and not all will be of sufficient quality. So, we also look at assessing the quality of such classifications using the simulated data and discuss a method for ranking them.

10.1 Classification Experiments

For these experiments the simulated data (*SIM*) is used so that we can quantify the error over the entire image and compare it with the estimated error from the test set. The training data used was generated as described in Section 3.3.

The classification techniques used were C4.5 (c4), back-propagation neural network (nn) and the instance based learner with one nearest neighbour (iblk1) and three nearest neighbours (iblk3). These classifiers were trained, used to classify the test set and then the entire image.

As we saw in the previous chapter, classification with all possible classes and a small number of attributes results in high error rates. We need to simplify the problem being tackled and improve the quality of the information being presented to each of the classifiers. One simplification is to firstly classify the high level classes in an image. Once we have a good classification of the image at this level we can look at classifying each class into its sub-classes to obtain a more detailed classification if required.

The 15 classes in the simulated image were grouped into three broad classes – `vegetation`, `urban` and `water`. The number of cases in each class can be seen in Table 10.1. As we are using the *SIM* dataset we know the classes of each pixel in the image, and the target classification can be seen in Figure 10.1.

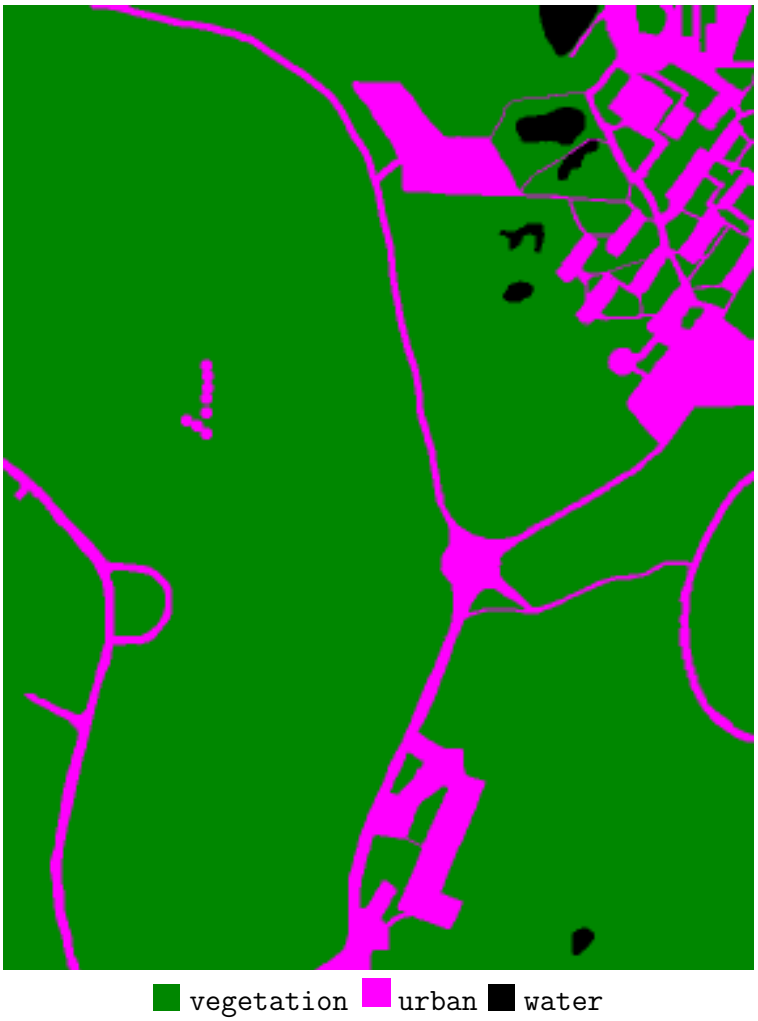


Figure 10.1: A three class classification of the simulated image.

Class	Training Set	Stop Set	Test Set	Image
vegetation	224	114	120	368649
urban	174	93	101	50873
water	28	13	17	3516

Table 10.1: Number of cases in each of the training sets used for a 3 class classification.

As discussed in Chapter 5 a large number of attributes can be generated in order to differentiate features in the data. The attributes used were generated from the original simulated spectral data and include an unsupervised AutoClass classification (as discussed in Section 2.4.11) and vegetation indices (as discussed in Section 5.2.3). The attributes generated were as follows.

band	original simulated spectral data, *band*=blu,grn,red,nir
pca_*i*	the principal components of the each spectral band *i*=1..4
sp_ac	AutoClass classification of the spectral data, each pixel from the image is used as a case in the classification and the spectral values for a given pixel from each of the simulated spectral bands are the attributes for that pixel
ratio_*ij*	ratio of pairs of spectral bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
dvi_*ij*	the difference of spectral bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
ndvi_*ij*	the normalised difference vegetation index of bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
tvi_*ij*	the transformed normalised difference vegetation index of bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
savi_*ij*_*c*	the soil adjusted vegetation index between bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$, with soil adjustment factor *c*=0.1,0.3,0.5,0.7,0.9
msavi_*ij*	the modified soil adjusted vegetation index between bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
sri_*ijk*	stress related index of bands *i*,*j*and *k*, *i*,*j*,*k*=blu,grn,red,nir, $i \neq j \neq k$

All attribute values were scaled to between 0 and 1. This was done by dividing each value by the maximum value across the entire image for that attribute. Target classes for each classification task were also mapped to values between 0 and 1.

In some cases, specifically for the vegetation indices, the values for a particular attribute were close to zero for all pixels in the training dataset after scaling. Even though these attributes had larger values over the entire image, they do not provide enough information to be used for training classifiers. Thus, the attributes that had values all in the range ± 0.0001 in the test set were discarded. This gave a total of 81 attributes for use in the classification.

According to the approach outlined in Chapter 4, the neural networks used had 81 inputs, corresponding to the 81 attributes used, 20 hidden nodes and 1 output. The outputs of the neural networks were assigned classes by dividing the output values into ranges such that both the number of false positives and false negatives were minimised.

Unless otherwise stated the neural network classifiers were set up as described in Chapter 4. The C4.5 and IBL classifications were as described in Section 2.4.

Each of the 81 attributes became the inputs to each of the classifiers and attribute selection was carried out. Typically around 10 attributes were chosen for each classifier, the minimum number of attributes chosen for a given classifier being one and the maximum being 18. Two examples of the attribute sets can be seen in Table 10.2.

A number of neural networks were trained on all attributes and contribution analysis was used for attribute selection. Extensive experimentation for the datasets discussed in this thesis found that removing attributes that had contributions greater than -0.2 and less than 0.2 consistently gave a set of relevant attributes. With an identified threshold we are able to completely automate the attribute selection process for neural networks[1].

10.2 Multi-class Classification

Firstly multi-class classifications with attribute selection for the three classification schemes were carried out.

In the case of the neural network classification the three classes were mapped to

[1] A threshold of ± 0.2 was appropriate for the datasets and types of classifications used here. However, the general applicability of this threshold to other domains and datasets would need to be investigated.

Multi-class C4.5	Binary NN water
sp_ac	blu
savi_blu_red_0.9	grn
	red
	nir
	pca_1
	pca_2
	pca_4
	dvi_blu_grn
	dvi_blu_red
	dvi_blu_nir
	dvi_grn_red
	dvi_grn_nir
	savi_blu_red_0.5
	savi_blu_red_0.7
	savi_blu_red_0.9
	savi_red_nir_0.3
	savi_red_nir_0.9

Table 10.2: Sample attribute sets after attribute selection.

values between 0 and 1. The output values of the network were then divided into three ranges to give 3 classes. The values used are shown in Table 10.3.

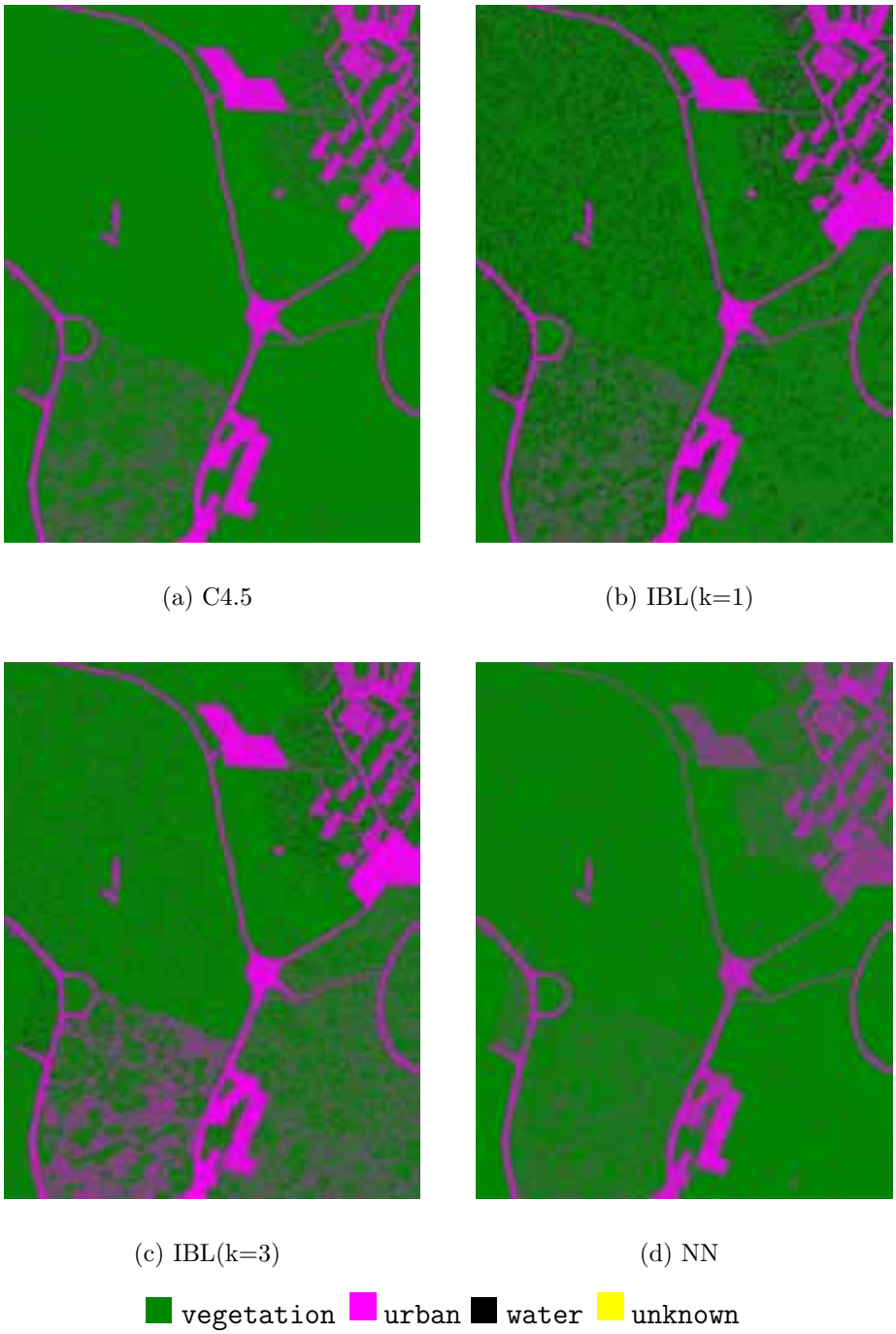
Class	Target Output	Output Range
vegetation	0.1	$\leq$0.3
urban	0.5	$>$0.4 and $\leq$0.7
water	0.7	$>$0.7

Table 10.3: Neural network target and output values.

The results of the four multi-class classifications can be seen in Table 10.4 and the classification of the entire image in Figure 10.2. In all cases the amount of `water` has been significantly overestimated, causing a large proportion of the speckle seen throughout the `vegetation` areas of each of the classified images.

Classifier	Test Error	Image Error
c4	15.1%	7.2%
iblk1	11.3%	15.4%
iblk3	13.0%	15.3%
nn	26.5%	10.1%

Table 10.4: Three class, multi-class classifications with attribute selection.



(a) C4.5 (b) IBL(k=1)

(c) IBL(k=3) (d) NN

Figure 10.2: Multi-class classifications.

10.3 Binary Classification

Next, binary classifications were carried out. For a given classification system a classifier was trained to recognise a single class, as described in Section 8.3.1. All cases in the class to be recognised from a single "in" class (target class label 0.9) and the cases for all remaining training classes are grouped into a "not in" class (target class label 0.1).

The results from the three binary classifications for each classifier were then combined using agreement classification as described in Section 8.5. The error rates for the agreement classifications generated can be seen in Table 10.5.

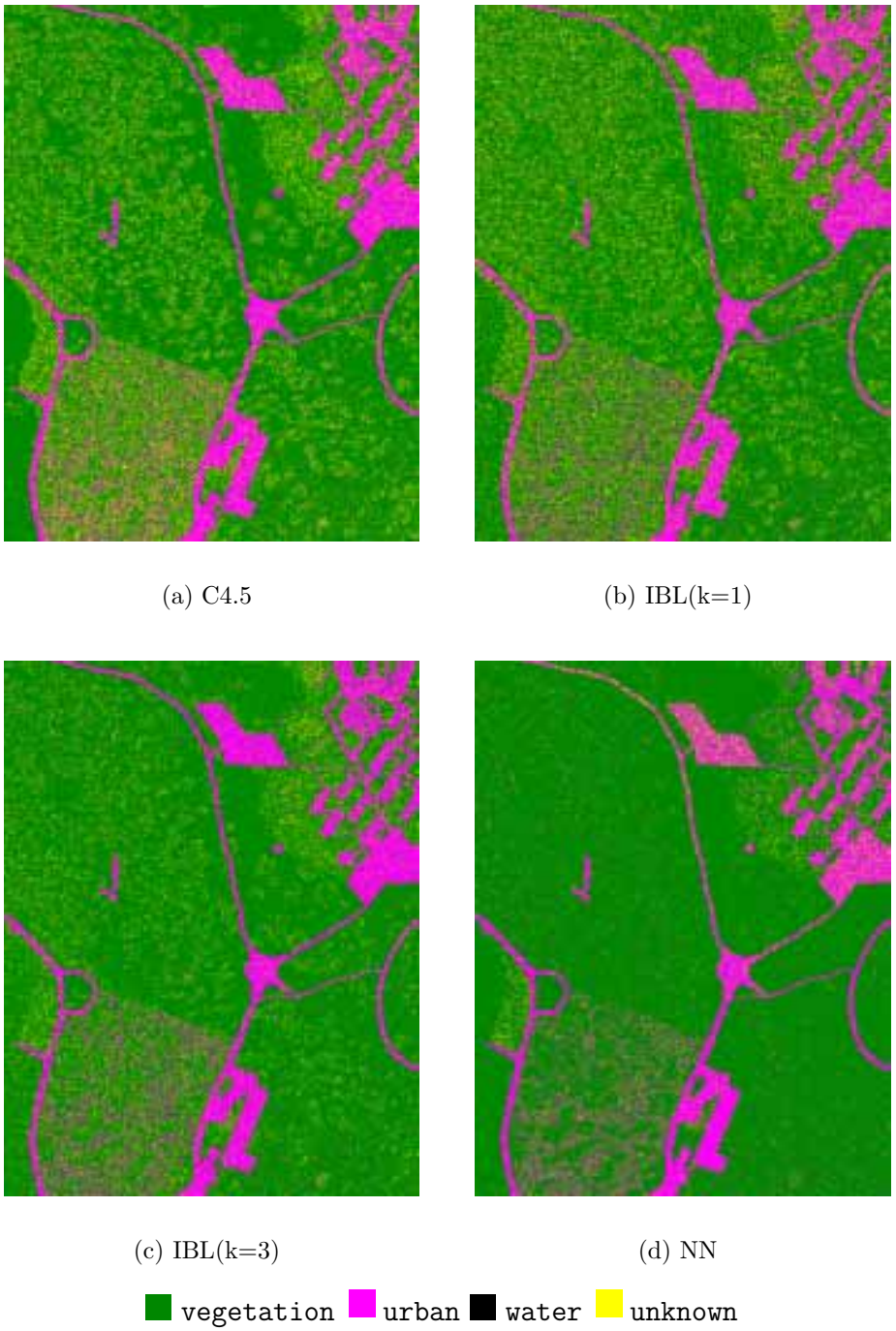
As discussed previously, once you start using agreement classification you have the ability to give pixels an `unknown` classification. The error given can be either the total number of pixels that have not been given the correct classification including those with an `unknown` classification (denoted as "Error" in the table) or the number of pixels that have been given a classification that is incorrect (denoted "Miscl Err" in the table).

	Test Error		Image Error		% Image
Classifier	Error	Miscl Err	Error	Miscl Err	Unclassified
c4	24.0%	7.6%	18.4%	5.2%	13.2%
iblk1	16.8%	9.2%	21.1%	7.1%	14.0%
iblk3	17.7%	10.5%	13.5%	5.4%	8.1%
nn	20.6%	14.3%	10.7%	6.1%	4.6%

Table 10.5: Three Class, binary classifications with attribute selection.

The overall error on the binary classifications is comparable to that on the multi-class classifications. However, we see a decrease in the number of misclassifications. That is, we should not consider the pixels that have been given an `unknown` classification to be true errors. These pixels have been identified as being different in some way from the members of each of the classes in the training dataset and require further investigation.

As can be seen in Figure 10.3, classification of the entire image shows that the pixels that have been given an `unknown` classification are pixels that tend to fall into the `tree` class (compare with the correct classification in Figure 9.4). A small number of isolated pixels have been given the `water` class but are largely in error.



(a) C4.5 (b) IBL(k=1)

(c) IBL(k=3) (d) NN

Figure 10.3: Binary classifications.

10.4 Multi-strategy Agreement Classification

Using the multi-class and binary classifications 38 additional agreement classifications were generated. Up to four of the multi-class or binary agreement classifications were combined using the agreement technique. These classifications are listed in Table 10.6 with their error rates. The combinations of the multi-class and binary classifications used in each agreement classification are denoted by a cross in the appropriate column. The multi-class and binary classifications have been included in the table for comparison.

The error rates for each of the classifications are generally within reasonable bounds on the test set, all being less than 40%. Again, we see a large reduction in the absolute error by considering only the pixels that have been misclassified to be true errors.

We also see that the error rates on the test set translates to a similar error rate for the entire image. However, we still need to identify the best classifications.

Binary Classifications				Multi-Class Classifications				Test Error		Image Error		%Image Uncl
c4	iblk1	iblk3	nn	c4	iblk1	iblk3	nn	%Err	%Miscl	%Err	%Miscl	
x								24.0	7.6	18.4	5.2	13.2
	x							16.8	9.2	21.1	7.1	14.0
		x						17.7	10.5	13.5	5.4	8.1
			x					20.6	14.3	10.7	6.1	4.6
				x				15.1	15.1	7.2	7.2	0.0
					x			11.3	11.3	15.4	15.4	0.0
						x		13.0	13.0	15.3	15.3	0.0
							x	26.5	26.5	10.1	10.1	0.0
x	x							30.3	4.2	30.0	3.0	27.1
x		x						27.3	5.5	23.7	3.1	20.6
x			x					29.8	6.3	21.2	3.4	17.9
	x	x						24.0	5.0	24.9	3.5	21.4
	x		x					29.4	5.5	25.1	3.0	22.0
		x	x					25.2	9.2	17.6	3.3	14.3
				x	x			18.9	5.9	17.8	4.3	13.5
				x		x		24.8	8.8	18.5	3.8	14.7
				x			x	29.0	11.8	13.6	3.5	10.1
					x	x		24.4	6.3	24.7	5.2	19.6
					x		x	29.8	7.1	21.5	3.3	18.2
						x	x	34.5	10.5	21.0	4.2	16.9
x	x	x						31.9	3.4	32.3	2.4	29.9
x	x		x					35.7	3.8	32.1	2.1	30.0
x		x	x					31.9	5.0	26.0	2.2	23.7
	x	x	x					31.5	5.0	27.9	2.3	25.5
				x	x	x		28.2	5.0	26.3	2.9	23.4
				x	x		x	31.5	5.5	22.9	2.4	20.5
				x		x	x	36.1	8.0	22.9	2.2	20.6
					x	x	x	36.6	4.6	29.2	2.3	26.9
				x	x	x	x	38.2	4.6	30.1	1.8	28.4
x	x	x	x					36.6	3.4	34.2	1.7	32.4
x				x				26.5	5.9	20.3	3.0	17.3
x					x			26.5	4.2	27.0	3.0	24.0
x						x		31.5	5.9	25.9	4.2	21.7
x							x	34.5	6.7	22.9	2.9	20.0
	x			x				23.1	6.3	22.9	3.1	19.8
	x				x			18.1	7.6	26.6	4.8	21.8
	x					x		29.0	5.0	28.5	4.1	24.3
	x						x	34.0	7.1	25.8	3.0	22.8
		x		x				21.0	10.1	15.4	3.5	11.9
		x			x			20.4	5.5	21.3	3.9	17.5
		x				x		26.1	7.6	21.1	4.4	16.8
		x					x	31.1	9.7	18.6	3.2	15.4
			x	x				23.1	10.5	13.0	3.5	9.5
			x		x			25.2	5.0	20.9	3.2	17.7
			x			x		28.2	8.4	19.7	4.7	15.0
			x				x	28.6	13.5	13.5	4.6	8.9

Table 10.6: Agreement classification error rates for *SIM* test set.

10.5 Kappa Evaluation

Choosing the classification with the smallest error on the test set is not necessarily the best classification. For example, the test set error on the binary iblk1 classification increased from 16.8% to 21.1% for the entire image. The C4.5 classification had an overall error that was low for the entire image, however a large number of `water` pixels were scattered throughout the `vegetation` areas. The image with the smallest percentage of pixels left unclassified may still contain many misclassifications, while the classification with the largest number of pixels left unclassified may be assigning an `unknown` class too readily.

Fitzgerald and Lees [54] also found that measuring the accuracy by the percent correct can be misleading. They showed that using the kappa statistic, which gives a measure of classification agreement, was far better for measuring the accuracy of classifications. The kappa statistic is increasingly used as a measure of accuracy, with examples of its use in [55, 54, 88, 134, 33, 100].

The kappa statistic is calculated from the error matrix of a classification. Given the error matrix for a k class classification:

		Actual Class				
		1	2	...	k	Total
Classed As	1	p_{11}	p_{12}	...	p_{1k}	$p_{1.}$
	2	p_{21}	p_{22}	...	p_{2k}	$p_{2.}$
	...			...		...
	k	p_{k1}	p_{k2}	...	p_{kk}	$p_{k.}$
Total		$p_{.1}$	$p_{.2}$	...	$p_{.k}$	1

where p_{ij} is the proportion of agreement between the test set and the classification given:

$$p_{ij} = \frac{p}{N}$$

where p is the number of pixels in class i classified as class j and N is the total number of pixels.

The Kappa statistic (κ) is then defined as follows:

$$p_o = \sum_{i=1}^{k} p_{ii}$$

$$p_e = \sum_{i=1}^{k} p_{i.} p_{.i}$$

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

When comparing the quality of classifications the one with the highest kappa value is the better classification.

The kappa statistic was used here to refine the evaluation of classified data by giving an agreement rating with the test set. Its use for this work was modified slightly to allow for the `unknown` classifications.

If the `unknown` class is labelled $i = 1$ we exclude it from the calculation and define kappa (κ) as follows:

$$p_o = \sum_{i=2}^{k} p_{ii}$$

$$p_e = \sum_{i=2}^{k} p_{i.} p_{.i}$$

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

Figure 10.4: The kappa statistic modified to ignore `unknown` classifications.

The kappa statistic gives us a further estimate of the accuracy of the classification than just using the overall percent correct.

10.6 Ranking Agreement Classifications Automatically

Automated ranking based on agreement with a target classification is a complete area of study on its own, and one for which there is no accepted approach that

can consistently provide reliable results [165].

The values of each of the error measures are highly correlated between the test set and the image, as seen in Table 10.7. That is, the measures of error on the test set are a reasonable estimate of the error over the entire image. The modified kappa value as calculated here takes into account the numbers of misclassified pixels, and ignores the unclassified pixels. So, it is possible to have an excellent match between the classified image pixels and the test set, but still have the majority of the image unclassified.

Error Measure	Correlation
kappa	0.994
error	0.975
misclassified pixels	0.898
unclassified pixels	0.983

Table 10.7: Correaltions between each of the error measures on the test set and over the entire image.

A heuristic method to automatically rank classifications was developed that aimed to maximise the modified kappa statistic and minimise the numbers of misclassified and unclassified pixels. The kappa statistic, as defined in Figure 10.4, measures the match with the target classification by removing the unclassified pixels, which means we can potentially get a very accurate classification by having the majority of the pixels unclassified. We therefore need to ensure that we choose classifications with the highest accuracy as well as the largest number of pixels classified as possible. So, the method used for measuring relative accuracy is given in Figure 10.5, where r is the rank given to a classification, u is the number of pixels given an *unknown* classification and N is the total number of pixels.

$$r = \kappa - \frac{u}{N}$$

Figure 10.5: The modified kappa statistic used for ranking classifications.

It is important to note that this produces a reasonable ranking of the classified images, not the optimal ranking. In the same way that the attribute selection techniques are heuristic techniques, this is also a heuristic technique for choosing the better classifications. Determining a more effective and accurate ranking

method is a substantial body of work in its own right [165] and so out of the scope of this work. None the less, this technique is suitable for our purposes.

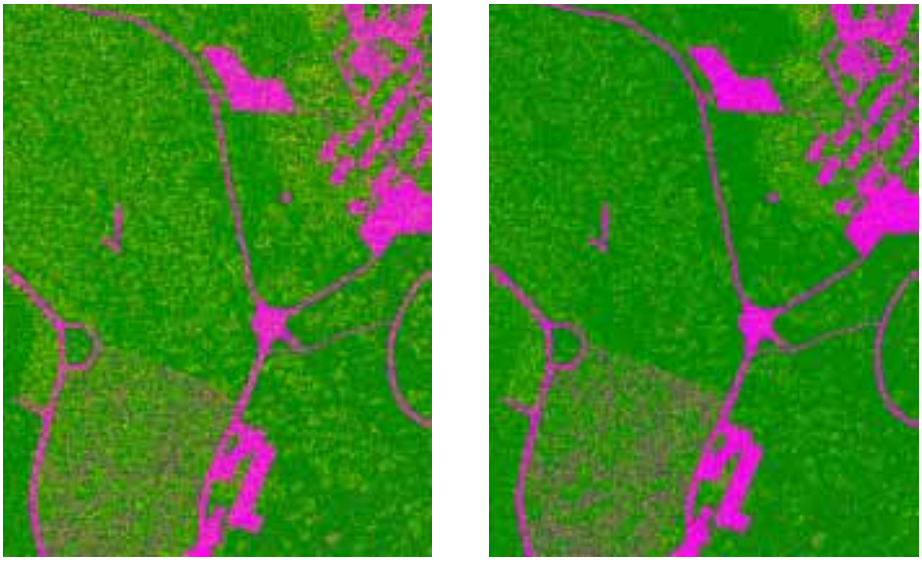
The rankings generated using this measure can be seen in Table 10.8. The multi-class classifications for each of the algorithms are ranked at the top, but still have a higher misclassification rate in general than for the agreement classifications. The agreement classifications also give more information about the areas of an image that need further investigation, that is, the areas of the image that can not be consistently classified. For this reason agreement classifications are chosen in preference, although the multi-class classifications have been included in Table 10.8 for reference.

Binary Classifications				Multi-Class Classifications				Test				Image			
c4	iblk1	iblk3	m	c4	iblk1	iblk3	m	Kappa	%Err	%Miscl	%Uncl	Kappa	%Err	%Miscl	%Uncl
	x							0.728	16.8	9.2	7.6	0.533	21.1	7.1	14.0
		x						0.702	17.7	10.5	7.2	0.624	13.5	5.4	8.1
	x				x			0.716	18.1	7.6	10.5	0.510	26.6	4.8	21.8
			x					0.643	20.6	14.3	6.3	0.641	10.7	6.1	4.6
				x	x			0.703	18.9	5.9	13.0	0.588	17.8	4.3	13.5
		x		x				0.661	21.0	10.1	10.9	0.616	15.4	3.5	11.9
		x			x			0.683	20.6	5.5	15.1	0.559	21.3	3.9	17.5
			x	x				0.632	23.1	10.5	12.6	0.632	13.0	3.5	9.5
	x			x				0.654	23.1	6.3	16.8	0.545	22.9	3.1	19.8
x								0.639	24.0	7.6	16.4	0.574	18.4	5.2	13.2
				x		x		0.626	24.8	8.8	15.9	0.582	18.5	3.8	14.7
					x	x		0.643	24.4	6.3	18.1	0.525	24.7	5.2	19.6
	x	x						0.649	24.0	5.0	18.9	0.527	24.9	3.5	21.4
		x	x					0.616	25.2	9.2	15.9	0.583	17.6	3.3	14.3
		x				x		0.618	26.1	7.6	18.4	0.554	21.1	4.4	16.8
			x		x			0.629	25.2	5.0	20.1	0.558	20.9	3.2	17.7
x				x				0.618	26.5	5.9	20.5	0.572	20.3	3.0	17.3
x					x			0.623	26.5	4.2	22.2	0.520	27.0	3.0	24.0
			x				x	0.551	28.6	13.5	15.1	0.595	13.5	4.6	8.9
x		x						0.611	27.3	5.5	21.9	0.541	23.7	3.1	20.6
			x			x		0.588	28.2	8.4	19.7	0.553	19.7	4.7	15.0
				x			x	0.558	29.0	11.8	17.2	0.602	13.6	3.5	10.1
				x	x	x		0.605	28.2	5.0	23.1	0.526	26.3	2.9	23.4
	x					x		0.598	29.0	5.0	23.9	0.497	28.5	4.1	24.3
	x		x					0.588	29.4	5.5	23.9	0.519	25.1	3.0	22.0
					x		x	0.569	29.8	7.1	22.7	0.536	21.5	3.3	18.2
		x					x	0.549	31.1	9.7	21.4	0.559	18.6	3.2	15.4
x	x							0.588	30.3	4.2	26.1	0.494	30.0	3.0	27.1
x			x					0.580	29.8	6.3	26.5	0.550	21.2	3.4	17.9
x						x		0.569	31.5	5.9	25.6	0.518	25.9	4.2	21.7
	x	x	x					0.569	31.5	5.0	26.4	0.504	27.9	2.3	25.5
				x	x		x	0.561	31.5	5.5	26.0	0.531	22.9	2.4	20.5
x		x	x					0.565	31.9	5.0	26.8	0.520	26.0	2.2	23.7
x	x	x						0.574	31.9	3.4	28.5	0.481	32.3	2.4	29.9
						x	x	0.513	34.5	10.5	23.9	0.530	21.0	4.2	16.9
	x						x	0.532	34.0	7.1	26.8	0.502	25.8	3.0	22.8
x							x	0.531	34.5	6.7	27.7	0.524	22.9	2.9	20.0
				x		x	x	0.510	36.1	8.0	28.1	0.529	22.9	2.2	20.6
x	x		x					0.536	35.7	3.8	31.9	0.477	32.1	2.1	30.0
					x	x	x	0.518	36.6	4.6	31.9	0.466	34.2	1.7	32.4
x	x	x	x					0.530	36.6	3.4	33.1	0.489	29.2	2.3	26.9
				x	x	x	x	0.505	38.2	4.6	33.6	0.485	30.1	1.8	28.4
					x			0.797	11.3	11.3	0.0	0.510	15.4	15.4	0.0
				x				0.718	15.1	15.1	0.0	0.704	7.2	7.2	0.0
						x		0.639	13.0	13.0	0.0	0.501	15.3	15.3	0.0
							x	0.490	26.5	26.5	0.0	0.554	10.1	10.1	0.0

Table 10.8: Automatic assessment of the *SIM* classifications using the test set rankings.

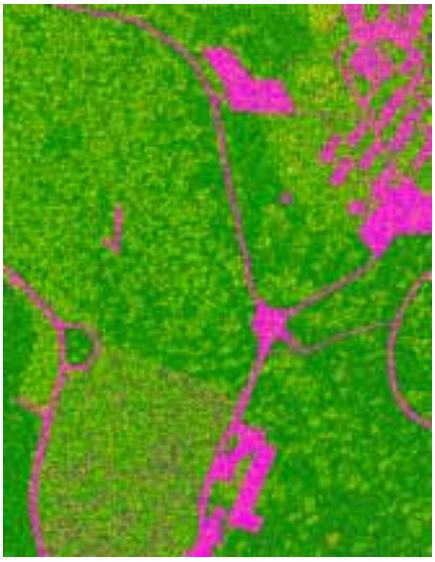
The best classifications are those that are either binary classifications or have no more than two classifications used. Using more than two classifications removes too much information from the image unnecessarily.

It must be noted that as a definitive ranking is difficult to give, the rankings here are merely indicative. In the same way that the error on the test set is only an estimate of the error on unseen data, the ranking method used is only an estimate of the relative merit of each classification – it will not always rank the best classification first. This method of ranking, however, is sufficient for removing the worst of the classifications.

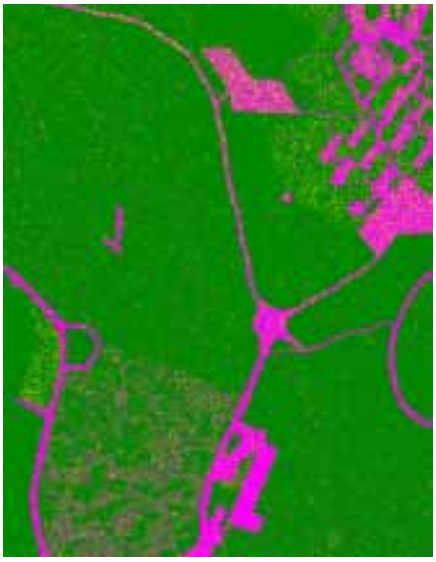


(a) Binary iblk1 classification.

(b) Binary iblk3 classification.



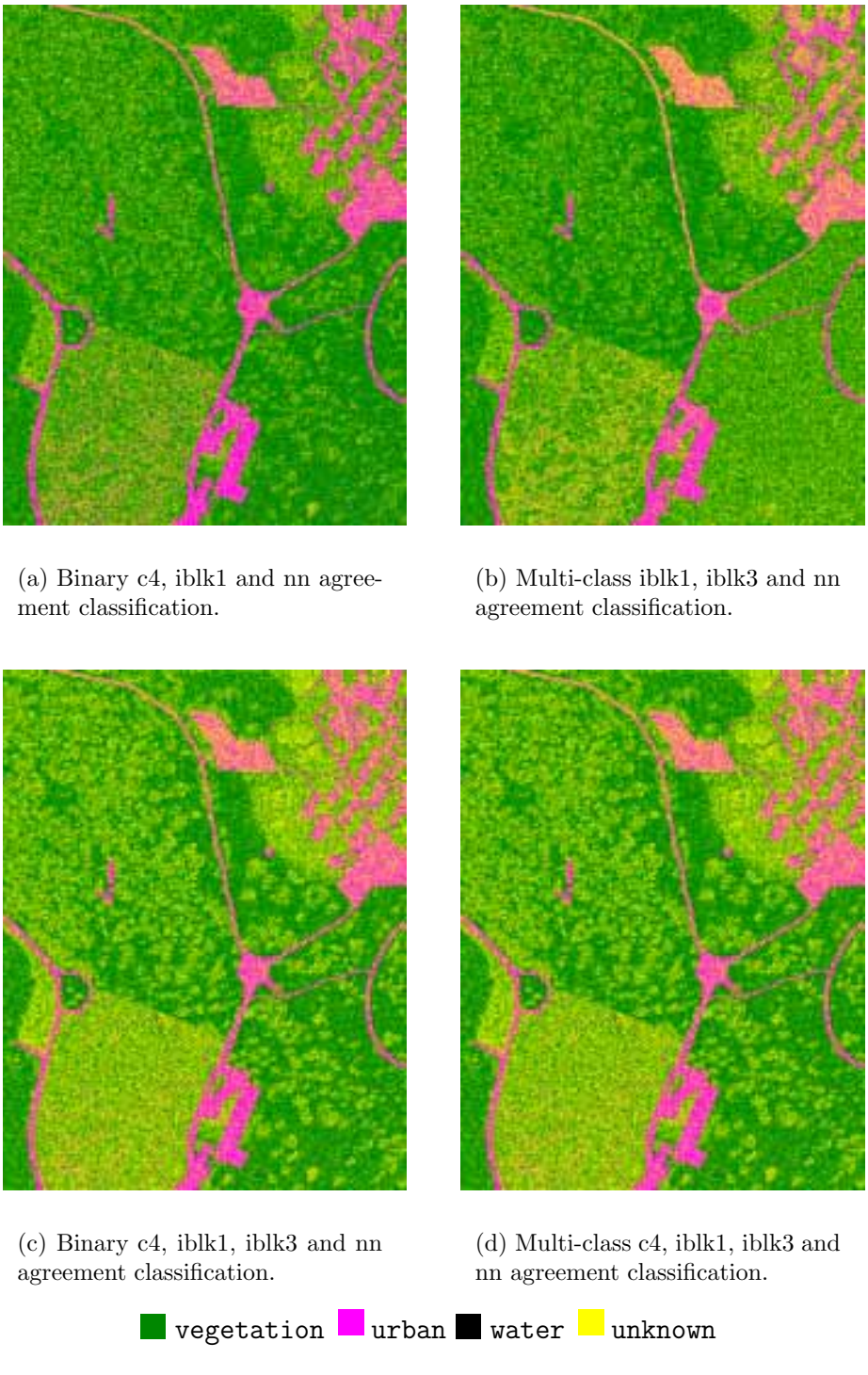
(c) Multi-class and binary iblk1 agreement classification.



(d) Binary nn classification.

vegetation urban water unknown

Figure 10.6: Top ranked *SIM* classifications.



(a) Binary c4, iblk1 and nn agreement classification.

(b) Multi-class iblk1, iblk3 and nn agreement classification.

(c) Binary c4, iblk1, iblk3 and nn agreement classification.

(d) Multi-class c4, iblk1, iblk3 and nn agreement classification.

Figure 10.7: Bottom ranked *SIM* classifications.

From the top ranked images the pattern of unclassified pixels is showing the confusion between grass, trees and urban structures. Areas of vegetation include both trees and grass which are quite different. There are areas of cultivated lawns and areas of weeds and uncultivated or grazed grasses that are clearly being confused. In addition, areas on the left side of the image have exposed areas of soil and rocks which are being confused with the urban features. The `water` class has been clearly missed in these classifications, and is not able to be reliably identified in any of the classifications. This is most likely due to the water plants covering the surface in the original image making the `water` class difficult to distinguish. In general this highlights the issue of distinguishing between different types of vegetation – that is, it is currently a difficult, if not impossible, task to carry out.

To do more detailed classifications, down to the level of tree species for example, you need to be able to reliably distinguish between features such as grass and trees. In these classifications we have made a start on making these distinctions. We could use these classifications to identify specific areas to collect more ground truth data and so clarify the differences between the different features in the image. The more data we have the better able we are to generate reliable classifications. In its absence we are still able to generate a reasonable classification with areas identified that we can give a reliable classification for.

10.7 Comparison with Maximum Likelihood Classification

In this section we compare the results of the agreement classifications with a traditional maximum likelihood classification. These will also be compared with more traditional and simplified C4.5, IBL and neural network classifications.

10.7.1 Multi-class Classification

The datasets as described in Section 10.1 were used to generate five additional classifications. However, for these experiments the only attributes used were the spectral data, that is the red, green, blue and near-infra red bands from the original simulated image.

All classifiers were configured as previously discussed, with the exception of the neural network. The topology of the neural network was modified to accommodate the reduced number of attributes. Each of the spectral values was mapped to an input node as usual. The number of output nodes was changed to three, one for each class, and three hidden nodes were used. The output values of each node were divided into 2 ranges - $\leq$0.5 and >0.5. The target output values for each class are shown in Table 10.9. The results of the five additional multi-class spectral classifications can be seen in Table 10.10.

Class	Target Output
vegetation	0.9 0.1 0.1
urban	0.1 0.9 0.1
water	0.1 0.1 0.9

Table 10.9: Neural network target and output values.

In all cases the error rates are much higher than for any of the agreement classifications generated using the spectral data and the generated attributes.

Classifier	Test Error	Image Error
maxlike	36.67%	53.82%
c4	14.71%	31.94%
iblk1	12.61%	41.26%
iblk3	15.13%	40.37%
nn	57.56%	87.97%

Table 10.10: Three class, multi-class classifications using only spectral data.

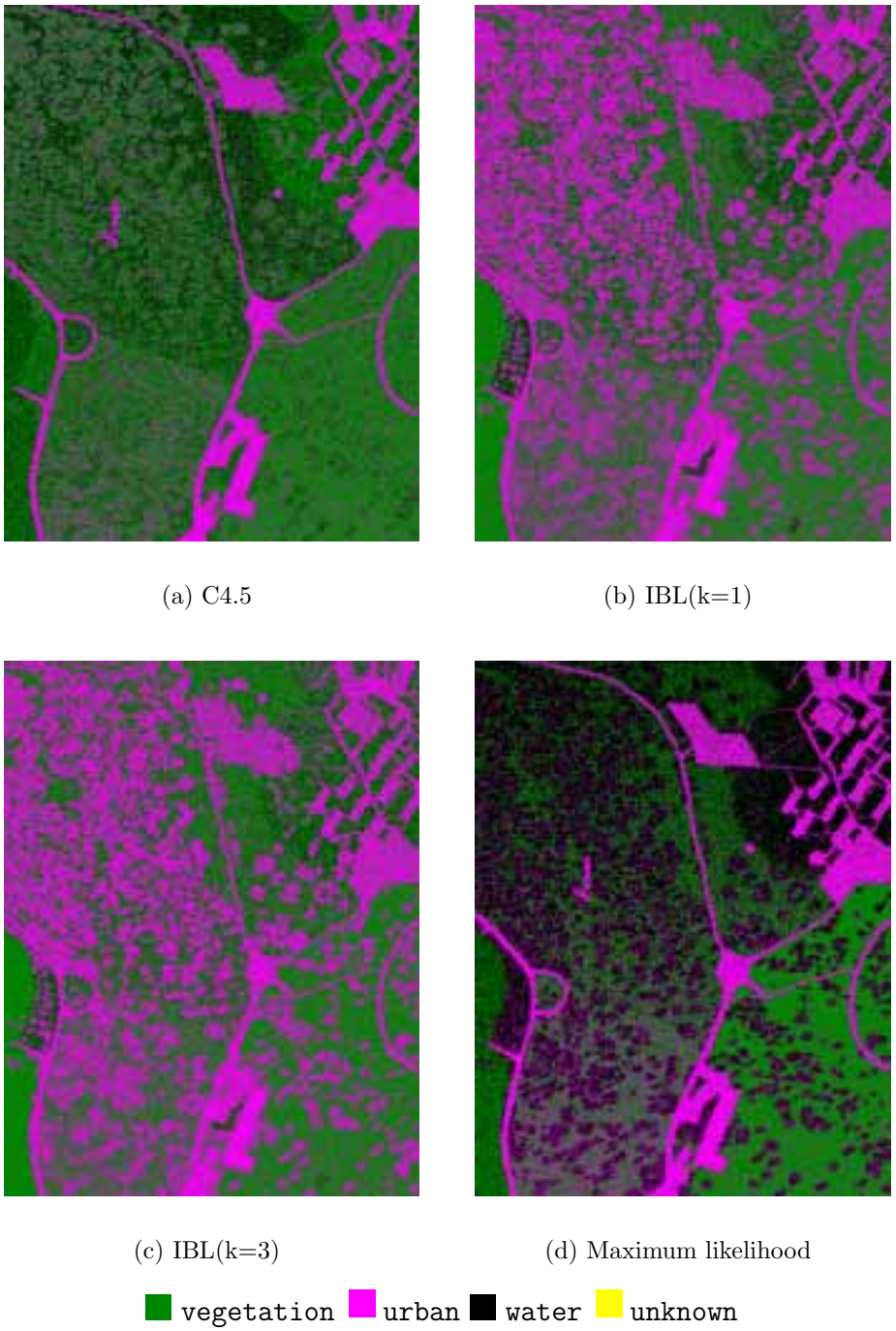
The neural network classification had the worst error rate because it gave everything a default `urban` classification. Varying the thresholds for the output values did not improve classification accuracy as all output values, for all classes, clustered together with no differentiation possible.

It might have been possible to find a network topology and configuration that could produce better results, although this has already been achieved with the agreement classifications previously discussed. However, as the main concern here is to compare maximum likelihood and agreement classification the multi-class neural network, trained using only the spectral data, has been excluded from further discussion.

The remainder of classifications that use only spectral data overestimate the num-

ber of `urban` or `water` pixels. The higher error rate for the maximum likelihood classification is a result of both being overestimated. The classified images are shown in Figure 10.8.

These classifications further support the widely accepted idea that additional data should be used in a classification of remotely sensed data. Not only that, classification can still be improved when the additional data is generated by pre-processing or pre-classifying the spectral data, as with the techniques discussed throughout this work.



(a) C4.5 (b) IBL(k=1)

(c) IBL(k=3) (d) Maximum likelihood

Figure 10.8: Multi-class classifications using spectral data attributes only.

10.7.2 Comparison Of Maximum Likelihood and Agreement Classification

The "best" and "worst" agreement classifications were compared with the spectral data only classifications. The "best" classifications were identified as those with the smallest error rate, smallest misclassification error or the smallest numbers of pixels with an `unknown` classification, or those with the largest kappa values, on either the test set or image data. Similarly, the "worst" were those that had the largest error rate, numbers of misclassifications or unclassified pixels, or the smallest kappa values. These classifications are listed in Table 10.11.

Id	Category	Binary Classifications				Multiclass Classifications			
		c4	ib1k1	ib1k3	nn	c4	ib1k1	ib1k3	nn
GD01	good		x						
GD02	good			x					
GD03	good				x				
GD04	good					x			
GD05	good						x		
GD06	good					x	x		
GD07	good	x	x	x					
GD08	good	x	x	x	x				
GD09	good		x			x			
GD10	good		x				x		
BD01	bad		x						
BD02	bad	x	x	x	x				
BD03	bad								x
BD04	bad					x	x	x	x
BD05	bad					x		x	x
BD06	bad						x	x	x
BD07	bad							x	x
BD08	bad	x							x
SP01	spectral data only					x			
SP02	spectral data only						x		
SP03	spectral data only							x	
SP04	spectral data only	Maximum Likelihood							

Table 10.11: Best and worst classifications.

The binary IBL (k=1) classification, is categorised as one of the best classifications (GD01) and one of the worst (BD01) as the kappa value on the test set is the highest, as well as the misclassification error being the highest of all the agreement

classifications. The agreement classification using the binary C4.5, IBL (k=1 and 3) and neural network classifications, is also in both categories (GD08 and BD02) as it has the smallest number of misclassified pixels but a large number of unclassified pixels.

The error rates, with 95% confidence intervals for the test set error, for the classifications can be seen in Table 10.12. As can be seen the maximum likelihood classification (SP04) is amongst the worst classifications by all measures, other than the percent unclassified. In particular, the error rate for the entire image is higher than any of the other classifications.

Id	Test			Image			Kappa	
	%Err	%Miscl	%Uncl	%Err	%Miscl	%Uncl	Test	Image
GD01	16.8±4.7	9.2	7.6	21.1	7.1	14.0	0.728	0.533
GD02	17.7±4.8	10.5	7.2	13.5	5.4	8.1	0.702	0.624
GD03	20.6±5.1	14.3	6.3	10.7	6.1	4.6	0.643	0.641
GD04	15.1±4.5	15.1	0.0	7.2	7.2	0.0	0.718	0.704
GD05	11.3±4.0	11.3	0.0	15.4	15.4	0.0	0.797	0.510
GD06	18.9±4.9	5.8	13.0	17.8	4.3	13.5	0.703	0.588
GD07	31.9±5.9	3.3	29.9	32.2	2.3	29.9	0.574	0.481
GD08	36.5±6.1	3.3	33.1	34.1	1.7	32.4	0.530	0.466
GD09	23.1±5.3	6.3	19.7	22.8	3.0	19.7	0.654	0.545
GD10	18.0±4.8	7.5	10.5	26.5	4.7	21.8	0.716	0.510
BD01	16.8±4.7	9.2	7.6	21.1	7.1	14.0	0.728	0.533
BD02	36.5±6.1	3.3	33.1	34.1	1.7	32.4	0.530	0.466
BD03	26.5±5.6	26.5	0.0	10.1	10.1	0.0	0.490	0.554
BD04	38.2±6.1	4.6	33.6	30.1	1.7	28.3	0.505	0.485
BD05	36.1±6.1	7.9	20.6	22.8	2.2	20.6	0.510	0.529
BD06	36.5±6.1	4.6	31.9	29.1	2.2	26.9	0.518	0.489
BD07	34.4±6.0	10.5	23.9	21.0	4.1	16.8	0.513	0.530
BD08	28.5±5.7	13.4	8.8	13.5	4.6	8.8	0.551	0.595
SP01	15.1±4.5	15.1	0.0	31.7	31.7	0.0	0.511	0.275
SP02	15.1±4.5	15.1	0.0	41.2	41.2	0.0	0.647	0.168
SP03	16.8±4.7	16.8	0.0	40.3	40.3	0.0	0.617	0.174
SP04	30.6±5.8	30.6	0.0	53.8	53.8	0.0	0.513	0.167

Table 10.12: Error rates for the best and worst classifications.

The kappa value on the test set for the maximum likelihood classification is not the smallest value, but it is similar to the kappa values for the worst of the agreement classifications. However, the kappa value is the smallest over the entire image by quite a large margin. Again, the low value of the modified kappa statistic over the entire image shows the clear inability of the classifier to generalise over unseen

data.

Landis and Koch [88] and Fitzgerald and Lees [54] ranked classifications based on their κ values as follows.

poor	$\kappa < 0.4$
good	$0.4 \leq \kappa \leq 0.75$
excellent	$\kappa > 0.75$

These ratings are meaningless for ranking the agreement classifications over the entire image as all get a rating of good. The ranges are not fine grained enough and do not distinguish even the worst of the classifications. However, the maximum likelihood classification falls well and truly in the poor category for the classification of the entire image.

The error for the agreement classifications over the entire image falls within or is close to the ranges given by the 95% confidence interval on the test set error. This is not the case for any of the spectral data only classifications. That is, agreement classifications, with the classifiers configured as described here, as gives us a more reliable estimate of error for unseen data.

Overall agreement classification provides a more consistent classification across different data or classifiers. Most importantly the test set error rates for agreement classifications are a more reliable estimate of the error over the entire image.

10.7.3 Overlap Between Classified Images

The overlap between each of the classifications listed in Table 10.12 were also compared. The kappa statistic, using the original definition from [54], was calculated between each pair of classifications, with the pairwise kappa values given in Table 10.13. This gives a measure of overlap between each pair of classifications. The kappa values for each classification on the test and image data sets have also been included for comparison purposes.

	Good Classifications									
	GD01	GD02	GD03	GD04	GD05	GD06	GD07	GD08	GD09	GD10
GD01	1.00									
GD02	0.74	1.00								
GD03	0.70	0.77	1.00							
GD04	0.75	0.83	0.78	1.00						
GD05	0.71	0.67	0.55	0.58	1.00					
GD06	0.67	0.69	0.62	0.69	0.69	1.00				
GD07	0.62	0.57	0.51	0.50	0.47	0.56	1.00			
GD08	0.58	0.53	0.51	0.48	0.45	0.53	0.86	1.00		
GD09	0.79	0.63	0.58	0.60	0.53	0.68	0.93	0.95	1.00	
GD10	0.77	0.60	0.54	0.54	0.60	0.68	0.82	0.81	0.80	1.00
BD01	1.00	0.65	0.58	0.57	0.56	0.67	0.62	0.58	0.79	0.77
BD02	0.58	0.53	0.51	0.48	0.45	0.53	0.86	1.00	0.62	0.61
BD03	0.59	0.66	0.80	0.59	0.42	0.66	0.74	0.84	0.70	0.64
BD04	0.52	0.53	0.53	0.51	0.51	0.61	0.59	0.62	0.58	0.60
BD05	0.55	0.59	0.59	0.57	0.46	0.61	0.64	0.67	0.63	0.59
BD06	0.53	0.55	0.54	0.50	0.52	0.61	0.60	0.63	0.58	0.61
BD07	0.57	0.61	0.64	0.54	0.46	0.59	0.66	0.69	0.62	0.60
BD08	0.63	0.69	0.83	0.62	0.48	0.68	0.76	0.86	0.72	0.67
SP01	0.30	0.30	0.27	0.27	0.29	0.32	0.34	0.32	0.32	0.33
SP02	0.20	0.19	0.18	0.16	0.19	0.19	0.24	0.23	0.20	0.22
SP03	0.21	0.19	0.19	0.17	0.19	0.20	0.26	0.25	0.21	0.23
SP04	**0.21**	**0.19**	**0.18**	**0.17**	**0.22**	**0.19**	**0.24**	**0.23**	**0.20**	**0.21**
Test	0.72	0.70	0.64	0.71	0.79	0.70	0.57	0.50	0.65	0.71
Image	0.53	0.62	0.64	0.70	0.51	0.58	0.48	0.46	0.54	0.51

(a) Kappa values between the "good" and other classifications.

	Bad Classifications								Spectral Classifications			
	BD01	BD02	BD03	BD04	BD05	BD06	BD07	BD08	SP01	SP02	SP03	SP04
BD01	1.00											
BD02	0.58	1.00										
BD03	0.59	0.84	1.00									
BD04	0.52	0.62	0.51	1.00								
BD05	0.55	0.67	0.57	0.71	1.00							
BD06	0.53	0.63	0.52	0.90	0.71	1.00						
BD07	0.57	0.69	0.62	0.65	0.82	0.68	1.00					
BD08	0.63	0.86	0.73	0.94	0.93	0.92	0.90	1.00				
SP01	0.30	0.32	0.18	0.27	0.24	0.26	0.22	0.24	1.00			
SP02	0.20	0.23	0.12	0.16	0.15	0.17	0.15	0.16	0.30	1.00		
SP03	0.21	0.25	0.11	0.17	0.15	0.17	0.15	0.16	0.28	0.75	1.00	
SP04	**0.21**	**0.23**	**0.14**	**0.16**	**0.15**	**0.16**	**0.15**	**0.17**	**0.29**	**0.32**	**0.35**	**1.00**
Test	0.72	0.53	0.49	0.50	0.51	0.51	0.51	0.55	0.51	0.51	0.64	0.61
Image	0.53	0.46	0.55	0.48	0.52	0.48	0.53	0.59	0.16	0.27	0.16	0.17

(b) Kappa values between the "bad" and spectral data only classifications.

Table 10.13: Kappa values between classifications.

The overlap between each pair of good classifications is higher than the overlap between the other classifications. This is indicated by the higher kappa values in Table 10.13(a). In particular, the overlap between the classifications using the spectral data only and all agreement classifications is substantially lower. That is, agreement and maximum likelihood classification result in substantially different classifications. In addition, the maximum likelihood classification has amongst the highest error rates, and the lowest kappa values for both the test set and the image data.

The average overlap kappa values for each category of classifications are given in Table 10.14. The average kappa values for the spectral data only classifications are significantly lower than for the agreement classifications. That is, there is a significant difference between the quality of the agreement classifications and the classifications that use only the spectral data.

	Good	Bad	Spectral
Good	0.659±0.126		
Bad	0.616±0.094	0.698±0.141	
Spectral	0.235±0.051	0.196±0.052	0.388±0.183

Table 10.14: The average kappa values for each group of classifications.

Table 10.15 shows the average kappa values across the test and image data sets. All classifiers give similar kappa values for the test set data ($\mu_{\kappa_{test}}$). However, the agreement classifications have a significantly higher kappa value for the image data ($\mu_{\kappa_{image}}$) and this value is consistent with the kappa value for the test set.

The spectral data only classifications are also less consistent in the error rates given. If we look at the variance in the kappa values, comparing each of the classifications with the image data, we see that it is significantly higher for the spectral data only classifications (see $\sigma_{\kappa_{image}}$ in Table 10.15). That is, the error rates using spectral data only, vary significantly and more than for the agreement classifications.

Overall, the maximum likelihood classification has the highest error for the image data, and overlaps least with the agreement classifications. Thus, maximum likelihood is clearly an inadequate technique for data of this kind. These results further support the findings of the Statlog Project [108] that traditional statistical techniques are not appropriate for use with remotely sensed data.

	$\mu_{\kappa_{test}}$	$\mu_{\kappa_{image}}$	$\sigma_{\kappa_{image}}$
Good	0.677±0.006	0.560±0.077	0.044
Bad	0.548±0.006	0.523±0.042	0.002
Spectral	0.572±0.004	0.196±0.053	0.143

Table 10.15: Average kappa values for each group of classifications on the test and image data.

In this case we only have the spectral data and the attributes generated from it – we do not have other ancillary data, such as climate or topology at our disposal. In spite of this we have still demonstrated the value of generating additional attributes from the available data – we can highlight information in the data and improve the accuracy of the classification.

10.8 Other Methods for Combining Classifications

Other methods of combining classifications were also investigated. Arbitration of classifiers and simple majority voting schemes were also investigated as an alternative to agreement classification.

Arbitration was carried out by using C4.5, IBL and neural network classifiers to determine the final class of a pixel, from the classifications given by the other trained classifiers. The target class for an arbitration classifier was the actual class for the given pixel. In all situations tested the classifications defaulted to that of the highest accuracy classifier. That is, arbitration did not give any additional information about the accuracy of a classification than a single classifier gave.

Similarly, majority voting schemes resulted in much higher misclassification rates as the conflicts between classifications were ignored. The value of agreement classification is predicated on conflicts indicating unreliable classification and so an `unknown` classification is preferable. Majority voting therefore did not result in any improvements in the number of misclassified pixels over the results obtained by agreement or arbitration.

When used in combination with the modified kappa statistic agreement classification produces higher accuracy classifications by reducing the number of mis-

classified pixels when compared with arbitration classification or majority voting schemes.

10.9 Discussion

In this chapter we demonstrated the combination of attribute generation, attribute selection and agreement classification techniques to generate classifications of remotely sensed images. The modified kappa statistic was also demonstrated for use in automatic evaluation. In particular, each step in the classification process has been automated. Traditional maximum likelihood classification was also compared with agreement classification, with agreement classification producing more consistent and accurate results.

The modified kappa statistic was shown to give us a reasonable method of ranking classifications. It is a better measure of agreement with the test set than the overall error, misclassification error or proportion of the image unclassified alone would provide. The best classifications were those that had the highest modified kappa and smallest number of unclassified pixels.

Once a relative ranking has been given to each of the classifications using the ranking value in Figure 10.5, we can choose the best classifications for investigation or further classification tasks. This allows us to generate a large number of classifications and discard the poor ones automatically.

We demonstrated the automation of classification using a relatively simple set of broad classes – vegetation, water and urban structures. The success of these techniques is not entirely due to the simplicity of the classification task. We have quantifiably demonstrated improvements in classification accuracy by comparison with more traditional maximum likelihood, neural network, C4.5 and IBL classifications. In addition to this, initial investigations carried out and published in [112], for distinguishing sub-classes within a given class using the same techniques have shown that reduced error rates can still be obtained.

It is possible that using agreement classification can reduce to that given by the worst classifier generated – the worst case scenario being that it gets minimal misclassified pixels because the majority of pixels are given an `unknown` classification. However, agreement classification is an improvement because we produce a number of alternative classifications and choose the best, rather than just generating a single classification from a single fixed classification method. With the number of classifications generated we can find a good classification, that is not necessarily the optimal solution, but is an improvement over other techniques.

Using agreement classification it is possible to maximise the identification of misclassifications. This, in combination with the use of the modified kappa statistic, ensures that the correspondance between the test data and the trained classifiers are as high as possible. Using the modified kappa statistic for ranking ensures that the worst classifiers do not bias the final results, and also that the number of misclassifications are not overestimated.

The number of classifications that can be used in an agreement classification are potentially unlimited. However, a point will be reached where no new information can be found in the data and the additional effort in training and classification will be of no benefit.

Another concern in this is that too many classifications, each misclassifying its own small subset of the data, will all disagree giving a mostly unclassified image. This was beginning to happen in classifications ranked lowest using the modified kappa statistic.

It could be argued that a brute force technique such as described here could benefit from exploratory analysis before, enabling a more reasoned argument about the attributes that are constructed for a given classification task. However, as discussed in Section 1.4, we argue that "mapping need not be an exercise in providing accurate maps, rather it can be used to identify where particular features might be found". The initial classification using the techniques discussed here can be an exercise in data mining.

Exploratory analysis can indeed be carried out, but it is important that information that individual classifiers might be able to use is not eliminated or overlooked. It is in fact recommended that the techniques discussed here are actively incorporated into the exploratory phase of classification. The attributes chosen, as well as the areas of an image that can not reliably be classified (i.e. are given an `unknown` classification) can be used to focus efforts by human experts in the exploratory phase.

As no clear patterns of specific attributes being favoured by a particular classifier or particular classification task emerged in the course of this work, it is believed that this is an appropriate data exploration technique. The fact that there were no such patterns also supports the notion that individual classifiers will identify different pieces of information in a given dataset, and that the information available in a given dataset will vary according to the quality and quantity of data.

A final benefit of the techniques discussed here for automatically generating and assessing classifications, are that they are general enough to be used with any number of different classification systems and datasets. The best combinations as judged by the modified kappa statistic will vary with the specific dataset and domain. Finding the optimal classification is impossible using any technique, however we can generate a set of good classifications for small, noisy datasets.

Chapter 11

Additional Case Studies

In previous chapters we have investigated numerous techniques that allow us to automatically generate and assess classifications of remotely sensed images more accurately. In this chapter we demonstrate the use of these techniques with the *CSU* and *RNP* datasets.

For both datasets a three class classification was carried out, attempting to identify `vegetation`, `urban` and `water` in each image. As before, the classifiers used were C4.5(c4), neural networks(nn) and IBL for one nearest neighbour(iblk1) and three nearest neighbours(iblk3).

As with the previous chapter, a large number of classifications were generated automatically and then ranked according to the error measures. The steps in classification are as follows:

1. Generate additional attributes using the techniques as described in Chapter 5.
2. Train binary and multi-class classifiers, and carry out attribute selection, using the wrapper method for C4.5 and IBL, and contribution analysis for the neural networks.
3. Train the classifier on the selected attributes using the training dataset.
4. Classify the entire image and give a ranking of each using the modified kappa value to give the correspondance between the classification and the test set, as described in Section 10.5.

For each classification system a multi-class and binary classification was generated, as described in Sections 10.2 and 10.3. From these classifications 38 agreement classifications were generated, as described in Section 10.4.

It is important to note that now were are classifying real images it is not possible to quantify the accuracy over the entire image, only that of the test set. We will need to rely to some extent on qualitative assessment of the classified images.

11.1 Charles Sturt University

The *CSU* dataset, introduced in Section 3.1, consists of a single ABVS image. It can be used to give us additional support in the use of the classification framework in that it can have a simple interpretation. The image is of the Charles Sturt University campus and has clearly identifiable areas of buildings and vegetation.

The total number of cases in the set are listed in Table 11.1 and were generated as described in Section 3.3.

class	training set	stop set	test set
vegetation	121	51	40
urban	64	16	23
water	74	20	24

Table 11.1: Number of cases in each of the training sets used.

The attributes used were generated from the original ABVS spectral data and include unsupervised AutoClass (Section 2.4.11) and maximum likelihood (Section 2.4.1) classifications, vegetation indices (Section 5.2.3), principle components analysis (Section 5.2.2) and the image pre-processing techniques (Section 5.2.1). The list of attributes used were as follows.

band	original ABVS spectral data, *band*=blu,grn,red,nir
*band*_pr	proportional spectral values of each ABVS band, that is, the total reflectance (R) is the sum of the reflectance values in each band (r_{band}), each pixel in *band*_pr is given the value $\frac{r_{band}}{R}$, *band*=blu,grn,red,nir
*band*_av_3x3	each pixel in this image is given the average value of all pixels in a 3x3 radius surrounding the given pixel, that is, the pixel in row r and column c in *band*_text_3x3 is given the value $\frac{\sum_{i=-1}^{1}\sum_{j=-1}^{1} r_{(r+i)(c+j)}}{9}$ where r_{ij} is the reflectance of the pixel in row i and column j in the given band, *band*=blu,grn,red,nir
pca_*i*	the principal components of the each spectral band i=1..4
sp_ac	AutoClass classification of the spectral data, each pixel from the image is used as a case in the classification and the spectral values for a given pixel from each of the ABVS bands are the attributes for that pixel
sp_ac_pr	AutoClass classification of the proportional spectral data, each pixel from the *band*_pr data, *band*=blu,grn,red,nir, is used as a case in the classification and the values for a given pixel from each of the *band*_pr images are the attributes for that pixel
ml_*c*	unsupervised maximum likelihood classifications for c =2,3,5,7,10,15,20,33,53 classes
*band*_3x3	a contextual AutoClass classification using the spectral data from the ABVS image, each pixel from the image is used as a case in the classification, the attributes for each pixel are the spectral value of the given pixel and the spectral values in a 3×3 radius around the that pixel for the given band, *band*=blu,grn,red,nir
*band*_pr_3x3	a contextual AutoClass classification using the proportional spectral data (*band*_pr), each pixel from the image is used as a case in the classification, the attributes for each pixel are the proportional spectral value of the given pixel and the proportional spectral values in a 3×3 radius around the that pixel for the given band, *band*=blu,grn,red,nir
ratio_*ij*	ratio of pairs of spectral bands$i$ and j, i,j=blu,grn,red,nir, $i \neq j$
dvi_*ij*	the difference of spectral bands i and j, i,j=blu,grn,red,nir, $i \neq j$

ndvi_*ij*	the normalised difference of bands i and j, i,j=blu,grn,red,nir, $i \neq j$
tvi_*ij*	the transformed normalised difference of bands i and j, i,j=blu,grn,red,nir, $i \neq j$
savi_*ij*_*c*	the soil adjusted vegetation index between bands i and j, i,j=blu,grn,red,nir, $i \neq j$, with soil adjustment factor c=0.1,0.3,0.5,0.7,0.9
msavi_*ij*	the modified soil adjusted vegetation index between bands i and j, i,j=blu,grn,red,nir, $i \neq j$
sri_*ijk*	stress related index of bands i,j and k, i,j,k=blu,grn,red,nir, $i \neq j \neq k$

All attributes were scaled to values between 0 and 1, using the range of values across the entire image. Target classes for each classification task were also mapped to values between 0 and 1.

In some cases, specifically for the vegetation indices, the values for a particular attribute were close to zero for all pixels in the training dataset after scaling. Even though these attributes had larger values over the entire image, they do not provide enough information to be used for training classifiers. Thus, the attributes that had values all in the range ± 0.0001 in the test set were not used, giving a total of 95 attributes.

Each of the 95 attributes became the inputs to each of the classifiers and attribute selection was carried out. The wrapper method of attribute selection was used for the c4, iblk1 and iblk3 classifications and contribution analysis was used for the nn classification. Typically around 6 attributes were chosen for each classifier, the minimum number of attributes chosen for a given classifier being one and the maximum being 23.

The outputs of the neural networks were assigned classes by thresholding the output values into ranges such that both the number of false positives and false negatives was minimised.

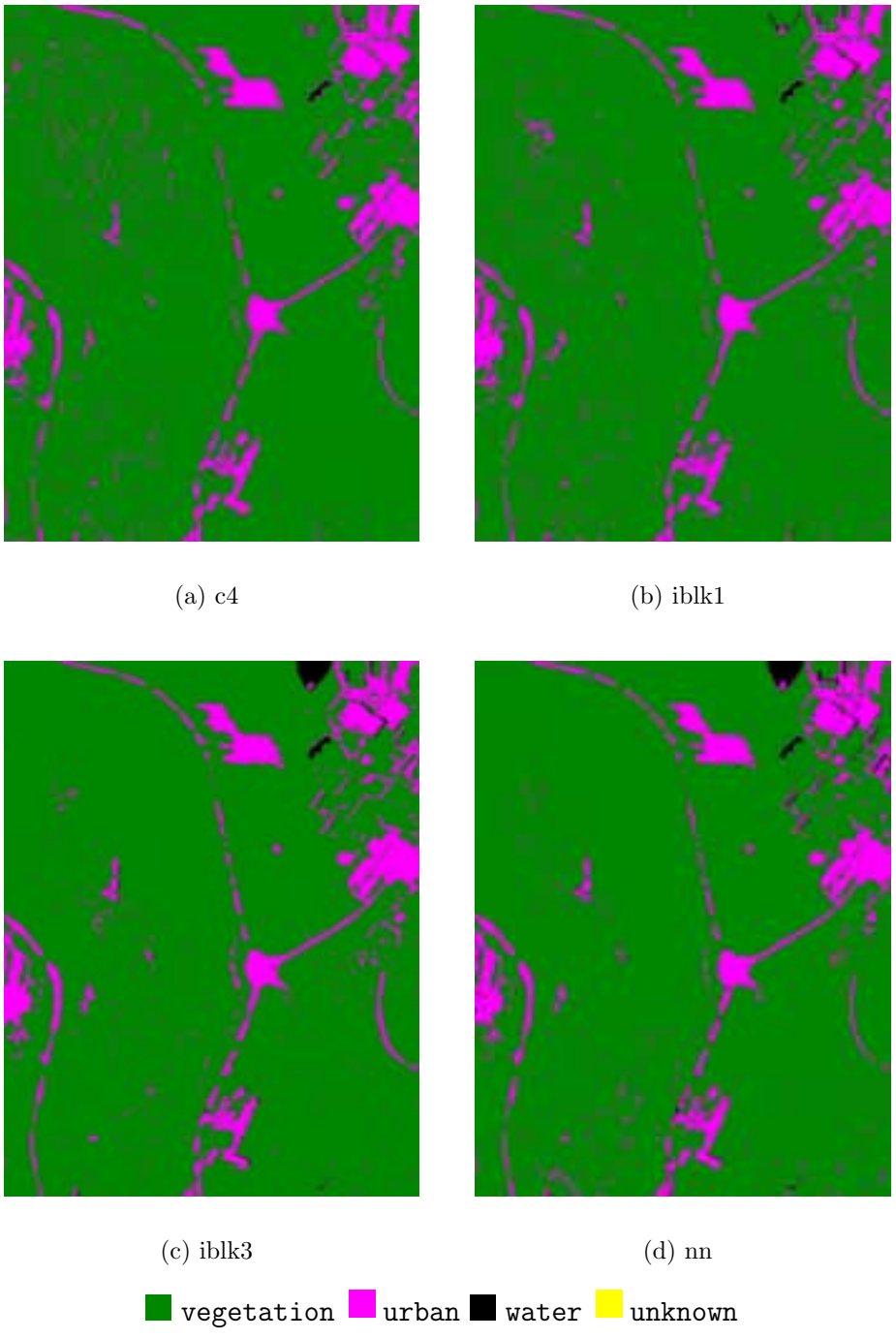
11.1.1 Multi-Class Classification

Firstly, the multi-class classifications with attribute selection were done for each of the classification systems. The results of the classification can be seen in Table 11.2.

Classifier	Test Error
c4	2.3%
iblk1	0%
iblk3	0%
nn	2.3%

Table 11.2: 3 class, multi-class classifications with attribute selection.

All classifiers have very low error rates on the test set and show a corresponding high quality classification over the entire image (see Figure 11.1). Not all of the lakes have been identified, but two have been identified reasonably accurately. However, in all classifications there has been an excessive assignment of water pixels (particularly in areas between buildings) and a large peppering of urban pixels in areas that are definitely vegetated. It would be desirable to remove, or at least to identify, these misclassifications.



(a) c4 (b) iblk1

(c) iblk3 (d) nn

Figure 11.1: *CSU* multi-class classifications.

11.1.2 Binary Classification

Next, binary classifications with attribute selection for the three classification schemes were carried out. Once again, as we are performing an agreement classification to combine the results of the binary classifiers we are able to assign an `unknown` class to pixels and remove some of the misclassifications[1]. The error rates and the percentage of the image unclassified can be seen in Table 11.3.

Classifier	Test Error	Test Miscl Err	% Image Unclassified
c4	19.5%	2.3%	23.45%
iblk1	0%	0%	4.69%
iblk3	1.1%	0%	3.77%
nn	6.9%	0%	6.39%

Table 11.3: Three class, binary classifications with attribute selection.

The overall error on the binary classifications is comparable to that on the multi-class classifications, with the exception of the C4.5 classification. In all cases we have reduced the number of misclassifications, as seen in the difference between the error and the misclassification error.

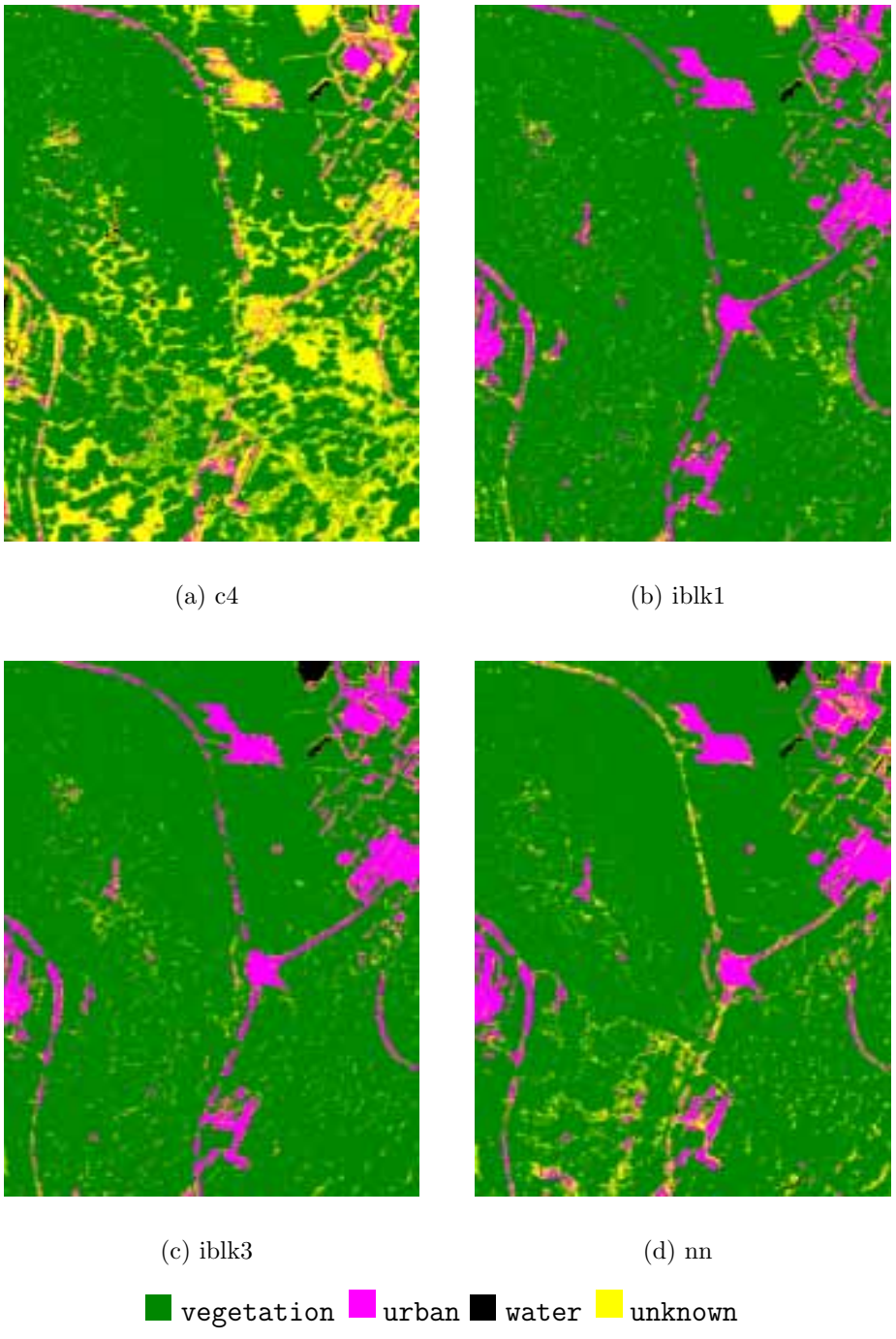
The classified images, as seen in Figure 11.2, show that the nearest neighbour classifications are slightly better, with the smallest number of errors and unclassified pixels.

The C4.5 classification (Figure 11.2(a)) still contains some `water` misclassifications, and most of the `urban` pixels have been given an `unknown` classification. However, it may be considered reasonably reliable if you are more interested in sub-classes of the `vegetation` class. A large number of the pixels that you would expect to fall broadly into the `vegetation` class have in fact been given `unknown` classification due to them containing dirt, rock, dry grasses or other such classes. This makes it a better quality classification than might at first be thought.

The neural network classification (Figure 11.2(d)) can also be considered a reasonably accurate classification. Two of the lakes have been identified, as well as part of the dam on the left hand side of the image. There are minimal misclassifi-

[1]As in Chapter 10 the error is the total percentage of pixels in the test set not given the correct classification, the misclassification error is the percentage of pixels given an incorrect classification but does not include those pixels given an `unknown` classification.

cations and again some of the under-story areas, particularly in the bottom right hand corner of the image, have been removed as atypical examples of `vegetation`.



(a) c4

(b) iblk1

(c) iblk3

(d) nn

Figure 11.2: *CSU* binary classifications.

11.1.3 Multi-Strategy Agreement Classification

Using the multi-class and binary classifications 38 additional agreement classifications were generated as described in Figure 8.5.

11.1.4 Automatic Assessment

As defined in Section 10.5, the modified kappa value for each classification was calculated. The modified kappa values for each of the classifications can be seen in Table 11.4, sorted from the highest modified kappa value to the lowest. The corresponding best four classifications for the entire image can be seen in Figure 11.3 and the worst four in Figure 11.4.

Binary Classifications				Multi-Class Classifications				Test Error			% Image
c4	iblk1	iblk3	nn	c4	iblk1	iblk3	nn	Kappa	Error	Miscl Err	Unclassified
					X	X		1.000	0.0	0.0	4.6
	X							1.000	0.0	0.0	4.7
		X						0.982	1.1	0.0	3.8
		X				X		0.982	1.2	0.0	4.5
	X					X		1.000	0.0	0.0	6.3
	X				X			1.000	0.0	0.0	6.6
						X	X	0.965	2.3	0.0	3.4
					X		X	0.965	2.3	0.0	4.5
		X			X			0.982	1.2	0.0	6.3
	X	X						0.982	1.2	0.0	7.5
				X			X	0.947	3.5	2.3	4.1
					X	X	X	0.965	2.3	0.0	6.2
				X	X			0.948	3.5	0.0	4.8
	X						X	0.965	2.3	0.0	6.8
				X		X		0.948	3.5	0.0	5.1
		X					X	0.948	3.5	0.0	5.4
				X		X	X	0.948	3.5	0.0	6.3
				X	X		X	0.948	3.5	0.0	6.7
				X	X	X		0.948	3.5	0.0	7.3
	X			X				0.948	3.5	0.0	7.7
				X	X	X	X	0.948	3.5	0.0	8.2
		X		X				0.932	4.6	0.0	6.7
			X					0.899	6.9	0.0	6.4
			X				X	0.899	6.9	0.0	7.6
			X			X		0.899	6.9	0.0	8.3
		X	X					0.899	6.9	0.0	10.1
			X		X			0.899	6.9	0.0	10.2
			X	X				0.883	8.1	0.0	9.8
	X		X					0.899	6.9	0.0	11.6
	X	X	X					0.899	6.9	0.0	12.9
X								0.736	19.5	2.3	23.5
X							X	0.736	19.5	2.3	25.1
X						X		0.740	19.5	0.0	25.6
X					X			0.740	19.5	0.0	25.8
X	X							0.740	19.5	0.0	26.5
X				X				0.724	20.7	2.3	25.3
X		X						0.726	20.7	0.0	26.3
X	X	X						0.726	20.7	0.0	27.6
X			X					0.698	23.0	0.0	28.1
X		X	X					0.698	23.0	0.0	29.1
X	X		X					0.698	23.0	0.0	29.6
X	X	X	X					0.698	23.0	0.0	30.2
						X		1.000	0.0	0.0	0.0
					X			1.000	0.0	0.0	0.0
							X	0.964	2.3	0.0	0.0
				X				0.947	2.3	0.0	0.0

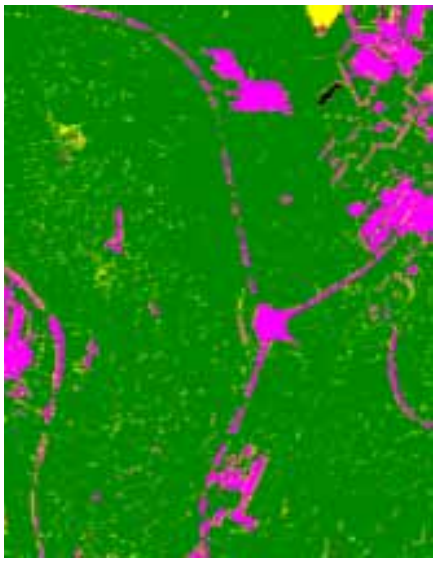
Table 11.4: Automatic assessment of the *CSU* classifications.

The agreement classifications that use the binary C4.5 classification have the highest error rates and pixels in the image left unclassified and so these classifications are all ranked at the bottom (examples can be seen in Figure 11.4). They are ranked low as they mostly only identify trees and grass. But even in this case it is due to pixels being given an `unknown` classification and the misclassifications on the test set are largely removed. All of the classifications that include the binary C4.5 classification have the lowest modified kappa ranking and so can be discarded.

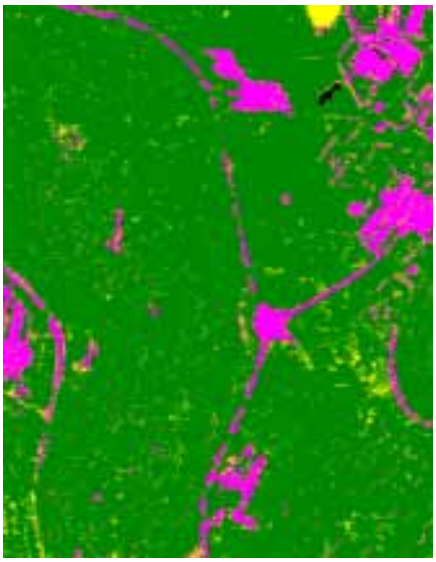
The multi-class and binary iblk1 and iblk3, as well as the agreement classifications between them were ranked highest. These are good classifications due to good identification of all three classes. The number of misclassifications has been clearly reduced – roads and urban features that have been misclassified have been given an `unknown` classification in the agreement classifications.

The multi-class neural network classification appears to be a reasonable classification (see Figure 11.1(d)) but the misclassification rates are much higher than for the agreement classifications that are ranked highest.

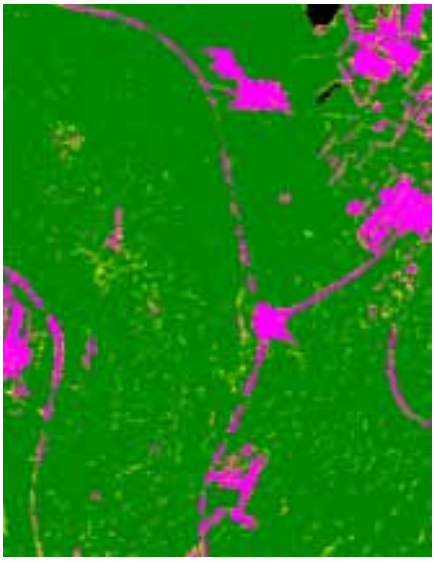
The highest ranked classifications are of high enough quality that they could be used to do a more detailed investigation of the image. As an example the vegetation components of the image could be isolated and more detailed classifications of the vegetation types could be done.



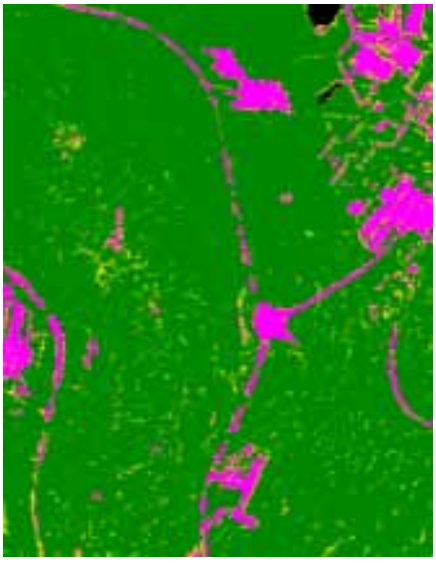
(a) Multi-Class iblk1 and iblk3 agreement classification.



(b) Binary iblk1 agreement classification.



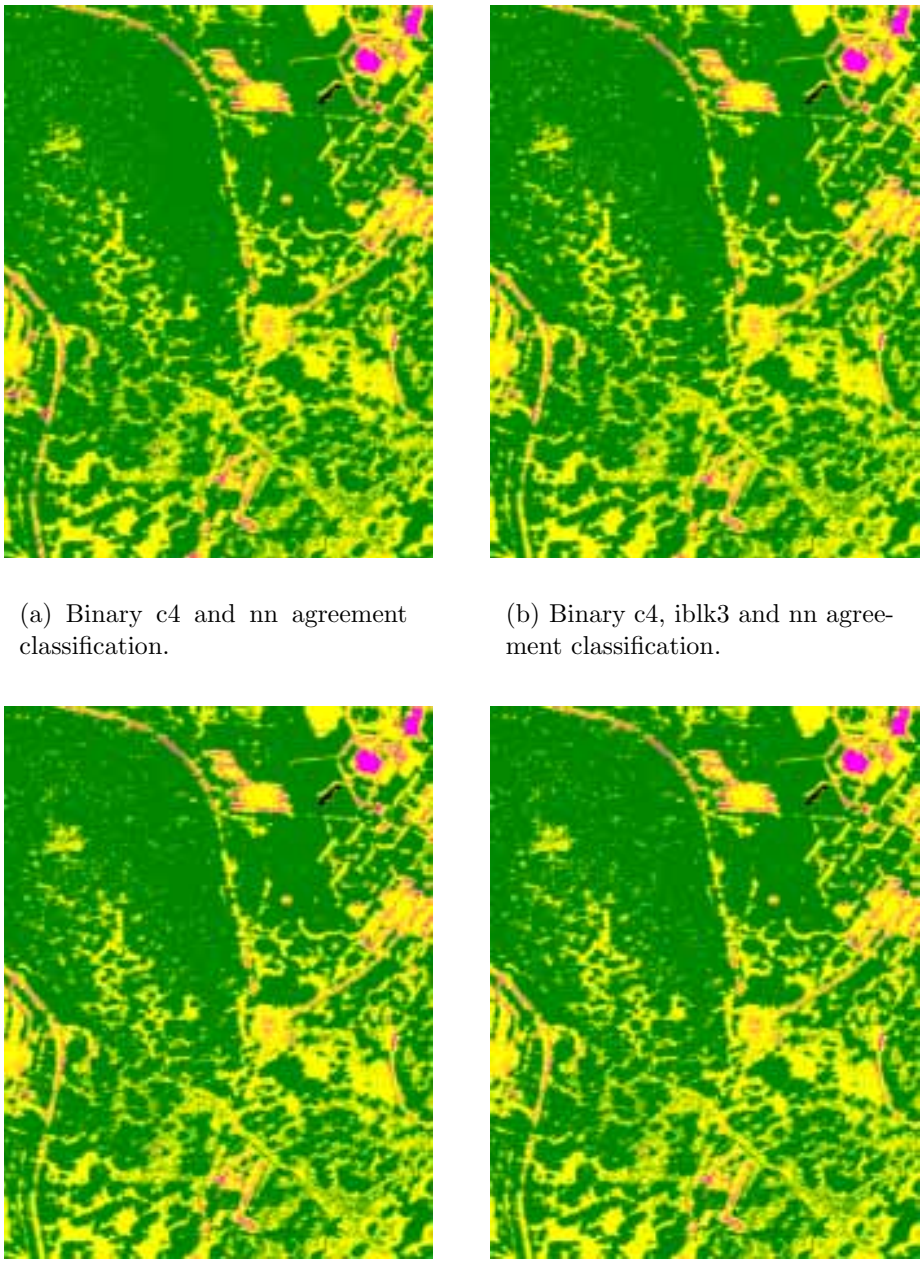
(c) Binary iblk3 agreement classification.



(d) Binary iblk1 and multi-class iblk3 agreement classification.

vegetation urban water unknown

Figure 11.3: Top ranked *CSU* classifications.



(a) Binary c4 and nn agreement classification.

(b) Binary c4, iblk3 and nn agreement classification.

(c) Binary c4, iblk1 and nn agreement classification.

(d) Binary c4, iblk1, iblk3 and nn agreement classification.

Figure 11.4: Bottom ranked *CSU* classifications.

11.2 Royal National Park

The *RNP* dataset, introduced in Section 3.2, consists of an ABVS image and three Landsat TM images.

Three class classifications were generated, the classes used and the number of cases in each set seen in Table 11.5.

Class	Training Set	Stop Set	Test Set
vegetation	65	43	85
urban	17	11	22
water	34	22	44

Table 11.5: Number of cases in each of the training sets used.

The attributes used were generated from the original ABVS and Landsat spectral data and include unsupervised AutoClass classification (Section 2.4.11) and principal components analysis (Section 5.2.2). Vegetation indices (Section 5.2.3) were not generated for the Landsat image as it is of a lower resolution than the ABVS image and the resulting number of additional attributes would have been too high to carry out attribute selection. The list of attributes used were as follows.

band	original ABVS spectral data, *band*=blu,grn,red,nir
may_tm$i$	Landsat TM spectral bands for May 1996, $i = 1..7$
aug_tm$i$	Landsat TM spectral bands for August 1996, $i = 1..7$
dec_tm$i$	Landsat TM spectral bands for December 1996, $i = 1..7$
*band*_pr	proportional spectral values of each ABVS band, that is, the total reflectance (R) is the sum of the reflectance values in each band (r_{band}), each pixel in *band*_pr is given the value $\frac{r_{band}}{R}$, *band*=blu,grn,red,nir
*band*_pr_3x3	a contextual AutoClass classification using the proportional spectral data (*band*_pr), each pixel from the image is used as a case in the classification, the attributes for each pixel are the proportional spectral value of the given pixel and the proportional spectral values in a 3×3 radius around the that pixel for the given band, *band*=blu,grn,red,nir
pca_i	the principal components of the each spectral band, i=1..4
pca_may_tm_i	the principal components of the May Landsat TM data, i=1..7
pca_aug_tm_i	the principal components of the August Landsat TM data, i=1..7
pca_dec_tm_i	the principal components of the December Landsat TM data, i=1..7
sp_ac	AutoClass classification of the spectral data, each pixel from the image is used as a case in the classification and the spectral values for a given pixel from each of the ABVS bands are the attributes for that pixel
sp_ac_pr	AutoClass classification of the proportional spectral data, each pixel from the *band*_pr data, *band*=blu,grn,red,nir, is used as a case in the classification and the values for a given pixel from each of the *band*_pr images are the attributes for that pixel
*band*_3x3	a contextual AutoClass classification using the spectral data from the ABVS image, each pixel from the image is used as a case in the classification, the attributes for each pixel are the spectral value of the given pixel and the spectral values in a 3×3 radius around the that pixel for the given band, *band*=blu,grn,red,nir

*band*_pr_3x3	a contextual AutoClass classification using the proportional spectral data (*band*_pr), each pixel from the image is used as a case in the classification, the attributes for each pixel are the proportional spectral value of the given pixel and the proportional spectral values in a 3×3 radius around the that pixel for the given band, *band*=blu,grn,red,nir
ratio_*ij*	ratio of pairs of spectral bands*i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
dvi_*ij*	the difference of spectral bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
ndvi_*ij*	the normalised difference of bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
tvi_*ij*	the transformed normalised difference of bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
savi_*ij*_*c*	the soil adjusted vegetation index between bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$, with soil adjustment factor *c*=0.1,0.3,0.5,0.7,0.9
msavi_*ij*	the modified soil adjusted vegetation index between bands *i* and *j*, *i*,*j*=blu,grn,red,nir, $i \neq j$
sri_*ijk*	stress related index of bands *i*,*j* and *k*, *i*,*j*,*k*=blu,grn,red,nir, $i \neq j \neq k$

All attributes were scaled to values between 0 and 1, using the range of values across the entire image. Target classes for each classification task were also mapped to values between 0 and 1.

In some cases, specifically for the vegetation indices, the values for a particular attribute were close to zero for all pixels in the training dataset after scaling. Even though these attributes had larger values over the entire image, they do not provide enough information to be used for training classifiers. Thus, the attributes that had values all in the range ± 0.0001 in the test set were not used, giving a total of 122 attributes.

Each of the 122 attributes became the inputs to each of the classifiers and attribute selection was carried out. The wrapper method of attribute selection was used for the c4, iblk1 and iblk3 classifications and contribution analysis was used for the nn classification. Typically around 8 attributes were chosen for each classifier, the minimum number of attributes chosen for a given classifier being two and the maximum being 21.

The outputs of the neural networks were assigned classes by dividing the output values into ranges such that both the number of false positives and false negatives was minimised.

11.2.1 Multi-class Classification

Firstly, the multi-class classifications with attribute selection were done for each of the classification schemes. The results of the classification can be seen in Table 11.6.

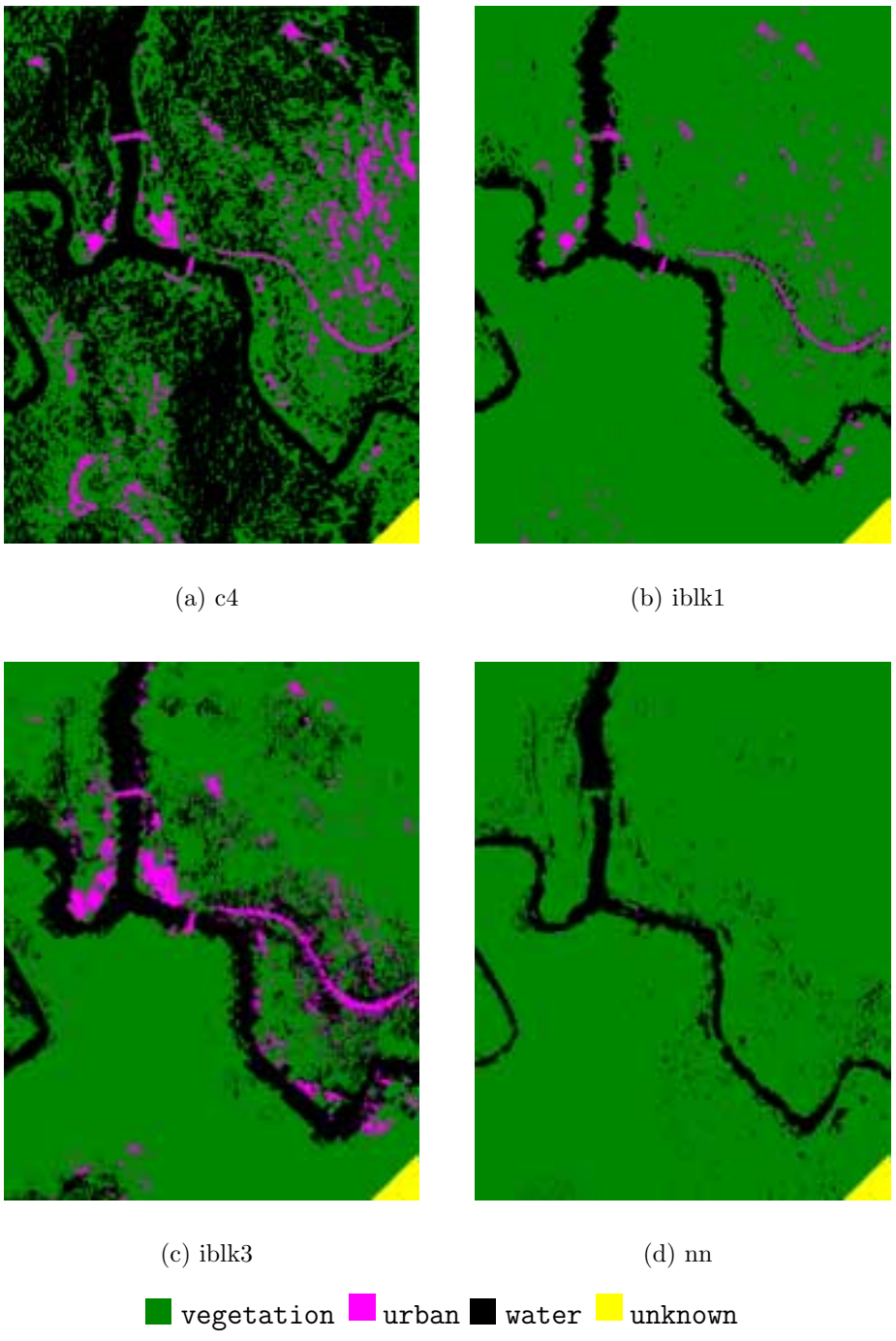
Classifier	Test Error
c4	11.3%
iblk1	1.9%
iblk3	2.6%
nn	17.9%

Table 11.6: 3 class, multi-class classifications with attribute selection.

The error rates for both the C4.5 and neural network classifications are considerably higher than for the IBL classifications. However, evaluation of the classification over the entire image shows that only the C4.5 classification contains a significant number of misclassifications (see Figure 11.5[2]). The neural network classification is still a reasonable classification in that there has been a reasonably clear distinction between the `vegetation` and `water` classes. However, there are a reasonable number of misclassified `water` pixels and no `urban` pixels have been identified.

Note, however, that the IBL classifications, in particular, are obviously influenced by the lower resolution of the Landsat data, in spite of the fact that all classifiers had at least one Landsat derived attribute chosen in the attribute selection process.

[2]Note that the bottom right hand corner of the image is missing due to missing data from the Landsat TM image.



(a) c4 (b) iblk1

(c) iblk3 (d) nn

Figure 11.5: *RNP* multi-class classifications.

11.2.2 Binary Classification

Next, binary classifications with attribute selection for the three classification systems were carried out. The error rates and the percentage of the image unclassified can be seen in Table 11.7.

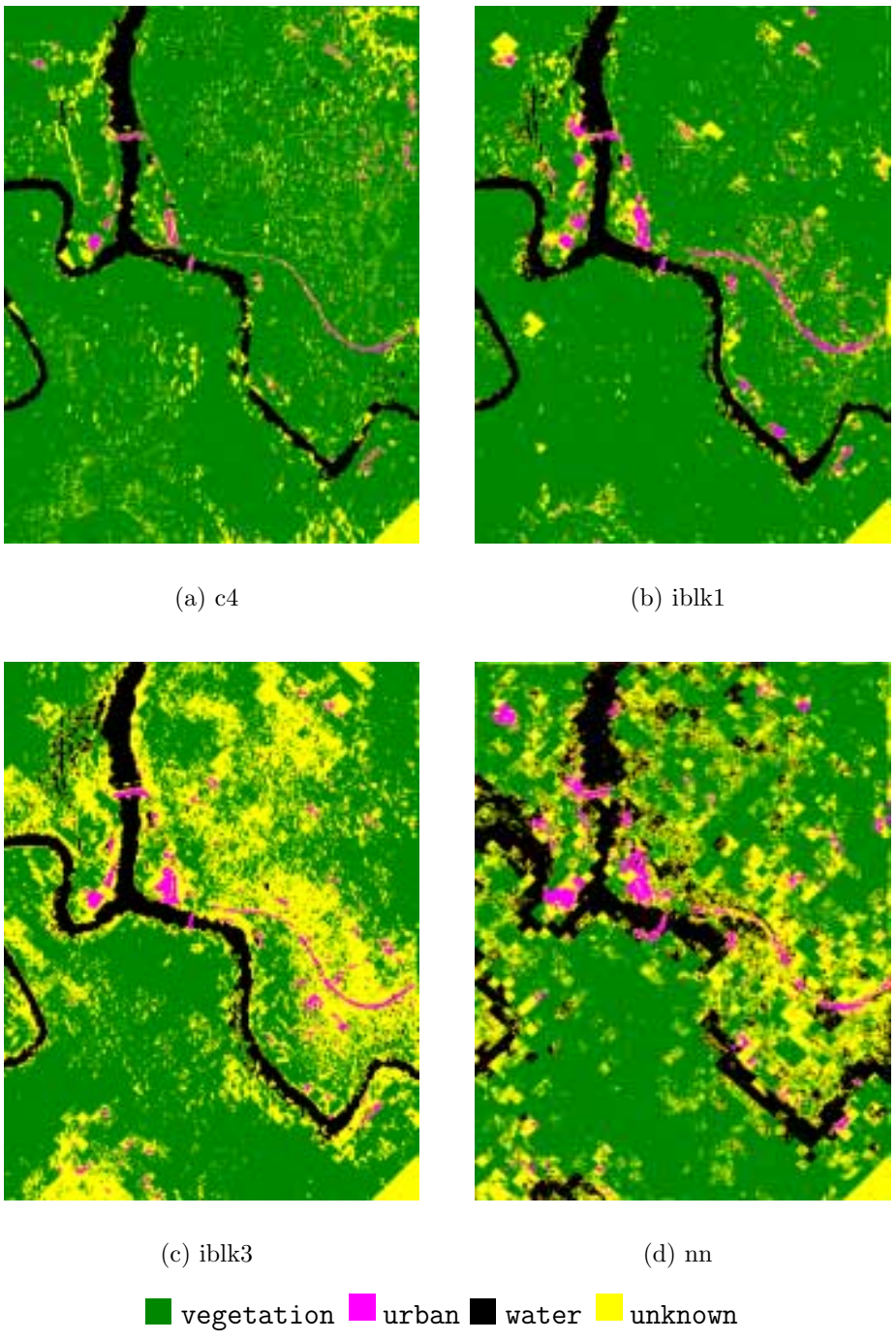
Classifier	Test Error	Test Miscl Err	% Image Unclassified
c4	13.2%	2.0%	9.31%
iblk1	5.30%	0.7%	7.98%
iblk3	15.9%	0.0%	31.46%
nn	19.2%	4.0%	25.20%

Table 11.7: Three class, binary classifications with attribute selection.

The overall error rate is comparable to that in the multi-class classifications, but we now see a large number of pixels being given `unknown` classifications for both the three nearest neighbour (iblk3) and neural network (nn) classifications. For the test set we again see a reduction in the number of misclassifications.

The classified images, as seen in Figure 11.6, show that the C4.5 classification is now slightly better. It is less influenced by the lower resolution of the Landsat data, but it also has minimal misclassified pixels (Figure 11.6(a)). The neural network classification (Figure 11.6(d)) has a large number of `water` misclassifications and is quite sensitive to the lower resolution Landsat data.

The poor performance of both the multi-class and binary neural networks may appear to indicate unsuitable network topologies. However, the results given do compare favourably to those reported in the literature for remotely sensed data (see Section 2.4.4 and Chapter 4). As previously noted, we failed to discover a topology with better classification accuracy for these data sets. In addition to this, the techniques described here supports the findings of Rogova [136]. That is, the results can be better for combined classifiers over individual classifiers and that this can mitigate the effects of a less than optimal network topology.



(a) c4 (b) iblk1

(c) iblk3 (d) nn

Figure 11.6: *RNP* binary classifications.

11.2.3 Multi-Strategy Agreement Classification

Using the multi-class and binary classifications 38 additional agreement classifications were generated as described in Section 8.5.

11.2.4 Automatic Assessment

As defined in Section 10.5, the modified kappa value for each classification was calculated. This ordering can be seen in Table 11.8, sorted from the highest modified kappa value to the lowest.

Binary Classifications				Multi-class Classifications				Test Error			% Image
c4	iblk1	iblk3	nn	c4	iblk1	iblk3	nn	Kappa	Error	Miscl Err	Unclassified
	X							0.914	5.30	0.70	7.98
	X				X			0.834	10.60	0.00	11.10
X								0.790	13.20	2.00	9.31
					X	X		0.807	12.58	2.65	16.19
X					X			0.748	16.56	0.66	13.29
					X		X	0.687	19.21	3.97	7.28
	X					X		0.815	12.58	0.00	20.64
X	X							0.769	15.23	0.00	16.82
	X						X	0.691	19.87	0.00	12.59
X							X	0.662	21.85	1.99	12.74
			X					0.730	19.20	4.00	25.20
						X	X	0.658	23.18	0.66	18.20
X						X		0.719	19.87	0.00	24.59
		X						0.779	15.90	0.00	31.46
					X	X	X	0.642	24.50	0.66	20.58
		X				X		0.757	17.88	0.00	36.01
		X			X			0.720	20.53	0.00	34.56
	X	X						0.731	19.87	0.00	36.16
			X			X		0.701	22.52	1.32	36.24
	X		X					0.701	22.52	0.00	40.32
			X		X			0.668	25.17	0.00	37.16
X		X						0.654	26.49	0.00	38.79
		X					X	0.600	29.80	0.00	35.33
X	X	X						0.639	27.81	0.00	40.97
			X				X	0.580	31.79	0.00	38.68
X			X					0.633	28.48	0.00	44.01
X	X		X					0.619	29.80	0.00	46.19
		X	X					0.665	26.49	0.00	51.68
				X	X			0.579	35.10	1.32	43.99
	X			X				0.608	32.45	0.00	47.15
				X		X		0.593	33.77	2.65	45.69
	X	X	X					0.636	29.14	0.00	53.36
X				X				0.552	37.75	0.00	48.44
				X	X	X		0.549	38.41	1.32	52.42
				X			X	0.476	44.37	0.66	45.23
X		X	X					0.575	35.10	0.00	55.94
X	X	X	X					0.567	35.76	0.00	56.93
		X		X				0.579	35.76	0.00	58.69
				X	X		X	0.460	46.36	0.00	48.05
			X	X				0.523	41.72	0.66	60.61
				X		X	X	0.444	48.34	0.00	54.01
				X	X	X	X	0.433	49.67	0.00	55.29
					X			0.870	1.90	0.00	0.00
						X		0.866	2.60	0.00	0.00
							X	0.679	17.90	0.00	0.00
				X				0.554	11.30	0.00	0.00

Table 11.8: Automatic assessment of the *RNP* classifications.

The overall error rates for these classifications are higher on average than seen on the *SIM* and *CSU* datasets. As the error rates for the multi-class and binary classifications are higher this is to be expected. We see the expected reduction in the number of misclassifications, which means a large number pixels with `unknown` classifications.

In spite of the fact that the modified kappa statistic is considered a better evaluation of classification accuracy we can see that it can still lead to poor choices being made when used for ranking. For example, the agreement classification between the binary iblk1 and multi-class nn classifications, as seen in Figure 11.7, is clearly a better classification than that for the binary nn classification. However, in the summarised table below we see that the binary nn agreement classification has a higher misclassification error and a larger proportion of pixels that have been left unclassified.

Binary Classifications				Multi-class Classifications				Test Error			% Image
c4	iblk1	iblk3	nn	c4	iblk1	iblk3	nn	Kappa	Error	Miscl Err	Unclassified
			X					0.730	19.20%	4.00%	25.20%
	X						X	0.691	19.87%	0.00%	12.59%

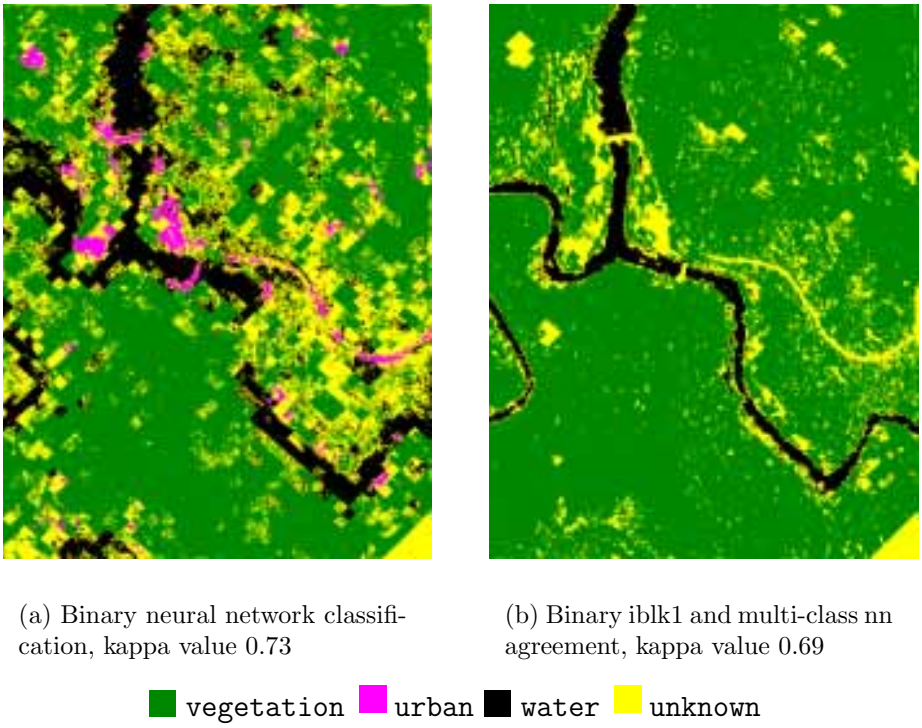
Once again, the method of ranking used is heuristic and so is not guaranteed to give the best overall classification the highest ranking.

For the top four classifications see Figure 11.8.

In this case the multi-class iblk1 and iblk3 classification (Figure 11.8(d)) in particular shows some impact from the lower resolution Landsat data, although these areas of the image have been left unclassified. Majority of the errors are reduced in the agreement classifications leaving a classification that could be considered reasonably reliable.

Agreement classifications using the multi-class C4.5 classification are given as the worst classifications. The modified kappa value for this classification is amongst the lowest. This is due to the significant overestimate of water giving a large number `unknown` classifications in the agreement classifications. See Figure 11.9 for the *worst* four classifications listed in Table 11.8.

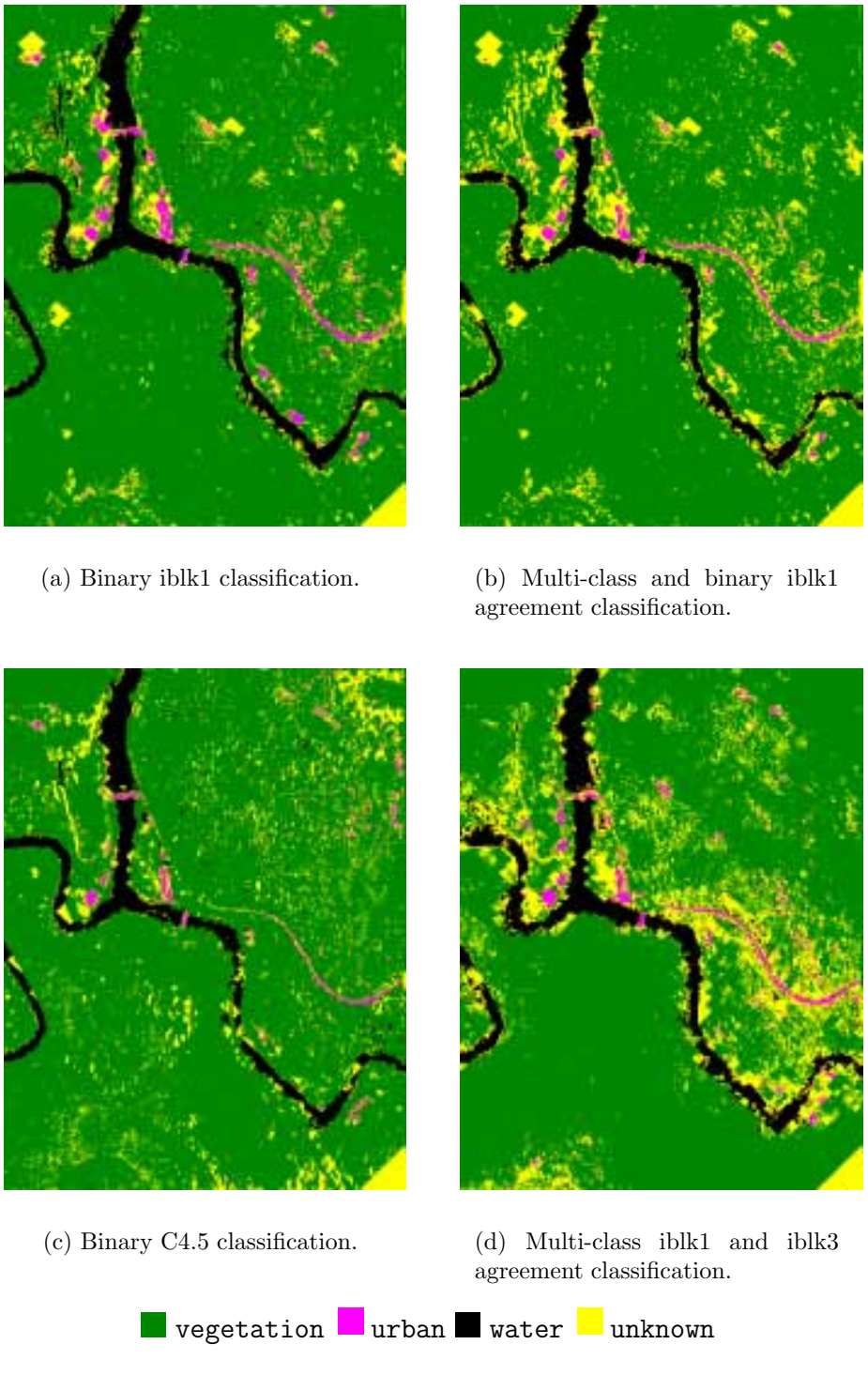
The multi-class neural network classifier is able to clearly distinguish between `vegetation` and `water`, without being affected by the lower resolution data. However, it does not recognise `urban` pixels and agreement classifications using it are ranked lower in the table.



(a) Binary neural network classification, kappa value 0.73

(b) Binary iblk1 and multi-class nn agreement, kappa value 0.69

Figure 11.7: Comparison of kappa values and classification quality.



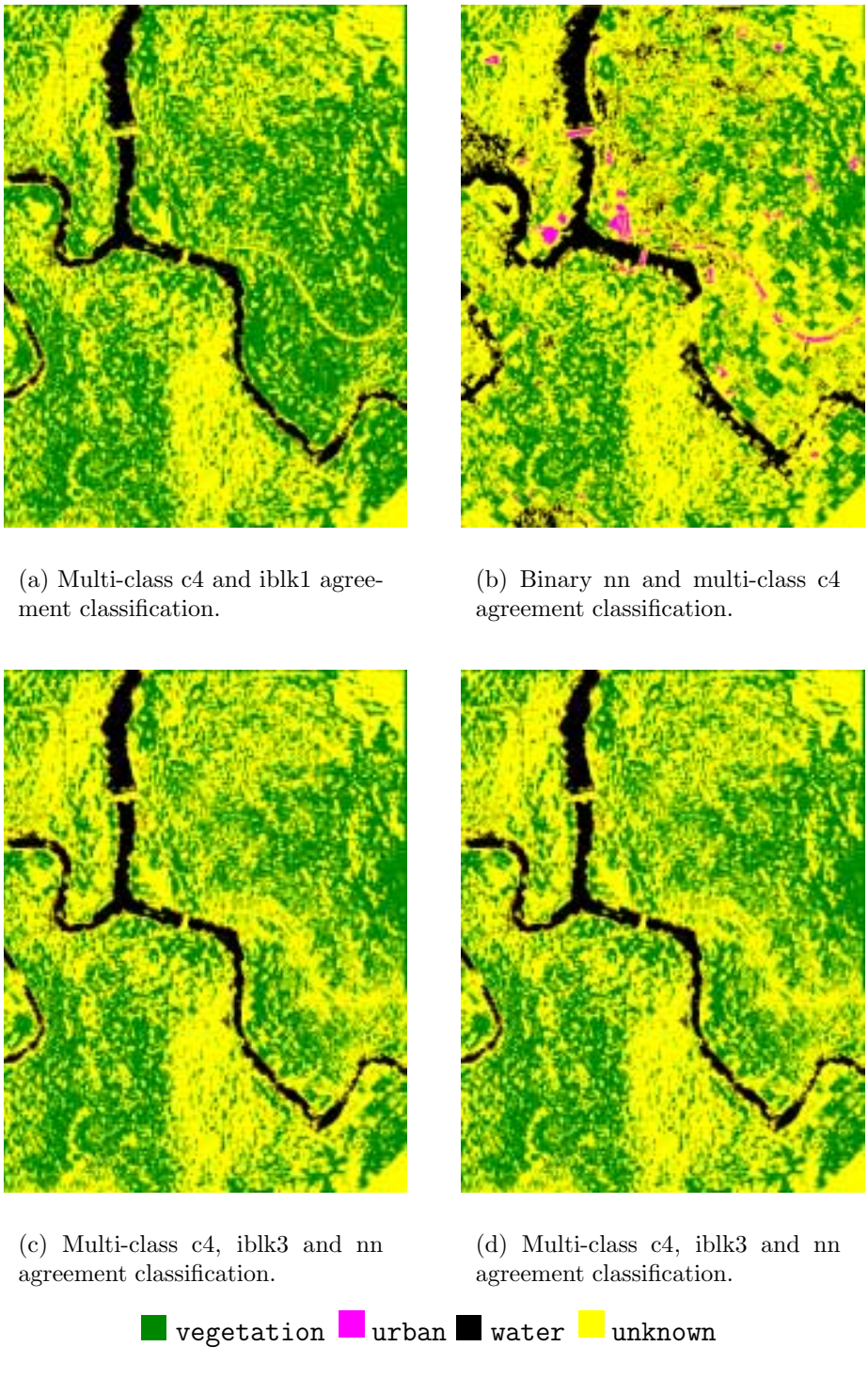
(a) Binary iblk1 classification.

(b) Multi-class and binary iblk1 agreement classification.

(c) Binary C4.5 classification.

(d) Multi-class iblk1 and iblk3 agreement classification.

Figure 11.8: Top ranked *RNP* classifications.



(a) Multi-class c4 and iblk1 agreement classification.

(b) Binary nn and multi-class c4 agreement classification.

(c) Multi-class c4, iblk3 and nn agreement classification.

(d) Multi-class c4, iblk3 and nn agreement classification.

Figure 11.9: Bottom ranked *RNP* classifications.

11.3 Discussion

In this chapter we have looked at classifying real remotely sensed images using the techniques described in previous chapters. We have demonstrated that it is possible to automatically generate and assess classifications.

For each of the datasets we generated large numbers of additional attributes to highlight information in the remotely sensed data. Both multi-class and binary classifiers were used to generate a number of classifications of the image. Attribute selection was carried out so that only a small number of relevant attributes was used for a given classification task. Finally, each of the classifications was automatically ranked according to its modified kappa value.

The classifications generated here serve to demonstrate that a single classification approach would not have produced consistent high quality classifications. In particular, the *SIM*, *CSU* and *RNP* datasets resulted in

- Vastly different subsets of attributes being chosen by each classifier trained, even the variation between classifiers generated for a single image were significant.
- Classifications for individual classifier algorithms varying quite substantially in quality across different images. For example the multi-class C4.5 classification for the *RNP* image was of a much poorer quality than for the *CSU* image.

By choosing a single classification method, with fixed attributes being used and single classification method, the quality of the classifications would not have been as high. By using the classifier to determine the most appropriate set of attributes, generating a range of classifications automatically and then ranking them allows us to improve the quality of the final classifications. In using this method of generating classifications we have removed the need to manually configure classifications for each new image and dataset obtained and are still able to achieve good quality maps.

A way to threshold rankings given to classifications so that the "bad" classifications can be discarded early in the process without discarding useful information would also be of value. This would also help to reduce classifier training times.

11.4 Conclusions

In this chapter we demonstrated the combination of the techniques discussed in this thesis for automatically generating an assessing classifications of remotely sensed data. Automated classification gives us the ability to generate a large number of alternative classifications and choose the best, which can then be used for mapping applications or use in further classifications and analysis.

Chapter 12

Conclusions and Future Work

In this thesis we have attempted to address some of the issues in automatically classifying remotely sensed data, however, there is still much work that can be done.

We have demonstrated a set of techniques that have the following features.

- We can produce more consistent results between the test set and the entire image. We can now be reasonably confident that the error on the test set will be reflected in the error over the entire image. That is, the test set error is a reasonable estimate of the image error.
- We can reduce the error in a classification by removing some of the misclassified pixels.
- Classifications can be generated and assessed automatically. The techniques used are general enough to be used with a wide range of classifier systems.

The key impediments to automating the classification of remotely sensed data that were overcome were:

- Large scale investigation of the available data to highlight the information contained in it by generating additional attributes and to use only the most relevant of those for a given classification task.

- Constraining the neural network classifier topology and training regime. Heuristics for determining when a given network has generalised the characteristics of classification task were given. An attribute selection technique for neural networks was introduced that is less computationally expensive than many of the currently used heuristic attribute selection techniques.
- Producing more consistent classification results between the training dataset and an entire image, and so enable automatic evaluation of the quality of the classifications generated.

12.1 Contribution Demonstrated in this Thesis

The key techniques demonstrated in this thesis are as follows.

Generating additional attributes for use in classification. A number of general techniques for highlighting information from the available data were discussed. These techniques were used to generate a large number of additional attributes for use in classification. (Chapter 5)

Neural network attribute selection and performance improvement. Neural networks, while obviously useful in pattern recognition problems, are not the classification panacea they are sometimes claimed to be. In particular, it was demonstrated that even though neural networks are able to distinguish objects in the presence of noise, they can be made to perform better when irrelevant or very noisy attributes are removed from the training data. An attribute selection technique that used the weights from a trained neural network to select the most relevant attributes was given. Heuristic methods for automating and improving the performance of neural networks were also given. (Chapters 4 and 7)

Simulating a remotely sensed image. Using the known properties of an actual image to generate an image where the cover type for each pixel is known. This allows us to compare classification techniques quantitatively. (Chapter 9)

Agreement classification to improve the reliability of a classification. A number of simple classifiers are trained on different views of the data. That

is, different subsets of the available data are used to train each classifier. A classification is given to a pixel only when all classifiers trained agree on class membership. The performance of individual classifiers is improved by using attribute selection. As well, agreement classification allows us to make effective use of small training datasets that may only contain data for a small number of the possible classes. (Chapter 8)

Automated generation and assessment of classifications Using the Kappa statistic, the misclassification error and the percentage of the image unclassified it is possible to rank the classifications generated. This ranking can be used to select the best classification or identify the top n classifications for further work or evaluation. (Chapter 10)

12.1.1 Neural Network Classification

Large variations in output values for a given case when classified by neural networks have been reported [147, 111]. This can be the case when networks have the same topology and are trained on the same data, with the only difference being the initial set of connection weights. This is due to the neural networks not being able to find consistent information in the training dataset in complex classification tasks.

Variations in output values were significantly reduced when the network topology is constrained, as described in Figure 4.1. The constraints limit the number of output classes and remove irrelevant information by attribute selection. The lessons learnt are that neural networks are still constrained by the information contained within the data.

In this work we demonstrated that the flexibility of neural network topology does not necessarily provide additional benefits. For the datasets used here it was possible to automate neural network classification using a reasonably constrained network topology and a well defined process for carrying out the classification task. Key points for automating classification are as follows.

- Limit the number of classes to be recognised to improve classification accuracy. This was particularly necessary in this work as attribute selection was only investigated for single output neural networks.

- Attribute selection using contribution analysis to limit the number of inputs to the neural network. This was automated by determining a minimum contribution that each attribute must make – the threshold used here of ± 0.2 was found to be appropriate for the datasets being investigated and was arrived at by extensive experimentation. Further work needs to be done on this to determine how to choose a reasonable threshold for a given dataset with a given number of input attributes.

- It is necessary to constrain the number of nodes in the hidden layers feeding into the single output node, for binary classification. This was achieved by using half the number of hidden nodes as there were input nodes. However, if the number of inputs was greater than 40 the number of hidden nodes defaulted to 20.

- The stopping point for the neural network training can be determined automatically by choosing the point of minimum error on the stopping set as described in Section 4.5.

- As most of the networks described here had only two classes to be recognised (0.1 and 0.9) a default thresholding of outputs of 0.5 could have been used. It was found, however, that the outputs typically did not separate that well and so a threshold was chosen that minimised both the number of false positives and the number of false negatives.

It was clearly demonstrated that the full capabilities and flexibility of neural network classification can be a weakness in certain types classification tasks.

12.2 Future Work

The central aim of this work was to allow the maximum amount of information to be extracted from the available data. In particular it provides a set of techniques that allows large scale automated searches for the most relevant attributes for a given classification task.

12.2.1 Generating Additional Attributes

Generation of attributes need not be limited to those used in this work. A wide range of alternative techniques are available that can be used to extract useful information from a given data set. In particular, there may be specific attributes that have already been proven to be useful in a given domain that can be used without much additional investigation. Another area that may provide additional techniques is that of data mining in large databases (for further discussion see [52, 126, 51]). The work described in this thesis provides the framework for using a wide range of such techniques.

It is important to note that just generating large numbers of attributes is unlikely to improve classification accuracy and that attribute selection will continue to be required. The aim of generating additional attributes is to provide alternative views of the data available and so ensure the maximum amount of information is extracted from it. From these alternative views of the data only the most relevant for the given classification task should be used. As we saw in the classification tasks carried out here, from an initially large number of possible attributes, only a relatively small number of attributes were actually chosen as being relevant and so used in a final classification.

Very large numbers of attributes are not necessarily needed before improvements in classification accuracy can be seen. Work published from this thesis, Milne [110], describes work that used attribute selection on only 17 attributes. As was the case in this work, irrelevant attributes may simply be those that have low signal to noise ratios and so are of no use in a classification task. The approach described in this thesis can easily be used on existing attributes without creating additional attributes. This will enable faster more accurate classifications by removing irrelevant attributes.

Trials carried out but not included here, tested attribute selection with up to 151 attributes. Unsurprisingly, this resulted in an increase in the training times but still gave improvements in the final error rates. However, it would not be surprising if increasing the number of attributes beyond this would quickly degrade classification accuracy due to sub-optimal subsets of attributes being chosen. This would, however, require further investigation.

If large numbers of attributes are to be investigated iterative attribute selection may be the solution. That is, dividing the available attributes into partitions and

then carrying out attribute selection on individual partitions. The smaller set of attributes chosen from each partition could then be either used in classification or further attribute selection carried out. More work would be required to find the best approach.

12.2.2 Neural Network Classification

One of the main criticisms of neural networks are that they are black box classifiers. That is, you can not reason about the information in the trained neural network. One of the aims of automated classification should be to also increase our understanding of remotely sensed data and in this case the natural environment. Work has already begun on interpreting the results from neural networks and is a necessary area of focus for this domain. Examples of interpreting neural networks can be seen in [36, 37, 141], but much work is still to be done in this area.

12.2.3 Simulating Remotely Sensed Data

While this work extended the work of [22] further improvements are still possible.

The values of neighbouring pixels will generally show some correlation and in this work we used a simple weighted sum to simulate this. Other approaches for correlating the values between neighbouring pixels should be investigated.

Masson and Pieczynski [101] found that variance of noise in images depends on the class. The noise that was added to the simulated image simply made the values a little bit smaller from one end of the image to the other. While this changed the statistical properties of the values slightly a more realistic method for adding noise to simulated images should be investigated.

Simulating remotely sensed data need not be restricted to evaluation of classifier systems. McKeowen et al. [104] discuss the use of large-scale virtual world databases for ground based simulations that incorporate remotely sensed data. Improving the simulation of remotely sensed data would mean that a variety of images could be generated to investigate a variety of scenarios. Such scenarios might include the simulation of pest damage or the effects of logging in forests. The simulated data generated for this work could be of used as the basis of creating

such simulations.

12.2.4 Attribute Selection

In spite of only having small noisy training sets we can highlight information in the available data to make the best use of it by generating additional attributes. We can then use attribute selection techniques to select the most relevant attributes for a given classification task.

As the attribute selection techniques described here are heuristic techniques, improvement in their performance is always going to be possible.

The attributes used in this work were limited to the spectral data and the attributes derived from it, however, they need not be limited to just these. Any additional information that can be used in a classification should be considered. Examples include soil maps, climate models, and expert knowledge.

The attributes chosen can give us insight into the characteristics of each class and the underlying properties that make classes different from each other. Analysis of the training dataset and the attributes selected will also ensure that artefacts in the data are not biasing the result. Investigations such as this would require input from domain expertise.

Machine learning techniques generally aim to produce a human readable form of the classifier to help extend our knowledge in a particular area. Further work needs to be done to extract more useful information from the attributes that are chosen as well as the classifiers that are trained from them.

12.2.5 Contribution Analysis

A number of techniques are available for attribute selection, one being the wrapper method as used in this work. However, this is a computationally expensive approach and is inappropriate for use with neural networks. For this reason contribution analysis was demonstrated for attribute selection in neural networks. This approach uses the weights from a trained neural network to determine the contribution of an attribute to the output of a neural network.

Attribute selection using contribution analysis was demonstrated and found to

improve the classification accuracy of neural networks in a number of different domains. The relevant attributes were determined by thresholding the contributions. More extensive investigation should be done, across a number of domains, to determine if there is an optimal threshold for a given domain based on the data or the number of attributes being used.

In this work the application of contribution analysis was only described for single output neural networks. This will need to be extended for networks with more than one output, and how best attribute selection can be done, needs to be investigated further.

12.2.6 Incorporating Contextual Information

While it is accepted that in classifying remotely sensed data the use of contextual, or spatial, information is important there are still no clear cut approaches for achieving this.

In this work contextual information was incorporated using quite simple approaches. In Section 5.2.1 method of generating an average image was outlined. This involved replacing each pixel in an image with the average value of it and its neighbours. In Section 5.2.6 we discussed using neighbouring pixels as attributes in unsupervised classifications.

Further work needs to be done to find more effective ways of incorporating contextual information either as attributes in a classification or within the actual classification system. Examples of this can be found in [101, 50, 48]. The classification framework described in this work is particularly suited to large scale investigation of this type as it is fully automated.

12.2.7 Agreement Classification

In this work we generated a large number of agreement classifications and then evaluated them using the kappa statistic. Clearly there were classifications that were generated of such a low accuracy that they would never improve the quality of a classification when using them in an agreement classification. Determining if such classifications can be discarded early on, and so also reduce the amount of computation required, should be investigated.

Combining the results from classifiers need not be as simple as described here and other methods should be investigated. For example, Drucker et al [43] used a two level classification scheme that involved training a classifier on a given task. The misclassified pixels were then used to train a second classifier and a third to arbitrate the results and give a final class. Such a technique could be adapted and extended to work for small noisy datasets and the classification framework described here.

12.2.8 Accuracy Assessment

While we demonstrated the use of the kappa statistic and the number of unclassified pixels to rank classifications much work can still to be done to refine this. For this work the kappa statistic gives the most significant measure of accuracy, but it needs to be adjusted to take into account the number of unclassified pixels in the final ranking.

Uebersax [165] discusses the complexity of determining level of agreement between two classifications and that this is an area requiring further research. In particular, for this work using the level of agreement to automatically rank classifications could be investigated further.

Ultimately the error on unseen data is always only an estimate based on classification accuracy for a classifier on a test set and so anything that improves this estimate will increase our ability to automatically rank classifications.

The kappa statistic could be used earlier in the classification process to remove less accurate information. This would also serve to reduce the processing requirements earlier. The focus of this work was in maximising the amount of information extracted from data and then only reducing this information at the very end. It would, however, be worthwhile investigating a reduction of the information available earlier in the classification process, and so potentially increasing the efficiency of the classification framework.

12.2.9 Increasing Classification Detail

Being able to generate more detailed classifications should be possible using the techniques described here. This could be to further refine class membership and

remove misclassified pixels, or to identify subclasses. As an example, for the classifications that were generated here we could take the `tree` class and further distinguish between trees, grass and water weeds, or to identify different tree species.

Initial investigations, carried out as part of this work and published in [111], segmented classes into sub-classes gave reasonable results. This will need to be investigated further, with extensive investigations now possible due to the automation of the classification framework.

12.3 Extensions to this Work

In addition to improvements that can be made to the techniques described in this work extensions into other areas are also desirable.

12.3.1 Knowledge-Based Systems

Incorporating remotely sensed data with geographic information systems has been hampered by the need for human interpretation and assistance [65]. That is, human expertise is still needed for tasks such as image registration and interpretation type tasks. By utilising expert knowledge as well means that we can extract further information from the data collected.

The problem with expert systems has been the human input for generating and maintaining the knowledge base. This issue has been addressed by Ripple Down Rules (RDR) expert systems [30, 31, 29]. A set of rules can be automatically generated using a training dataset and the expert can make changes to the rules, only in the given context, when errors are found.

While RDR still requires human input it can, at least, be partially automated and it is the ideal platform for incorporating expert knowledge into a classification. That is, once the knowledge base has been generated it can be used to generate classifications for use in agreement classifications.

It may also be possible to use an RDR type approach to improve agreement classifications and automated ranking of classifications, for example. In the case of agreement classifications a set of rules might be generated that take into account

properties of the data as well as the classifications generated. For the error estimation a simple set of rules for determining a good classification versus a bad classification might be generated.

12.3.2 Maintaining Classification Systems

Transience or concept drift is also a major concern. The objects in remotely sensed data are always changing and the classifications generated must evolve and adapt. Pest or fire damage, for example, are problems that always have the potential to exist but will not occur at predetermined times. Such problems will mean a different context is required for mapping these areas from remotely sensed images, and at some point in the future that context would once again become meaningless. How we deal with these types of issues is a major area of research on its own.

Much work has been done in remote sensing to detect changes between remotely sensed images collected at different times. For examples see [55, 123, 142]. Being able to detect changes requires that you can accurately generate classifications from different images that can be compared. Once this can be achieved changes will become easy to track. However, to reduce the amount of work done in classifications it would be better to be able to update the classification system rather than have to classify from scratch.

Cheon and Chang [25] used multiple neural networks to identify transient problems in nuclear power plants. A number of classifiers trained in my way mean that a large number of concepts can potentially be identified by training a large number of classifiers. The binary classification scheme then serves to identify the most appropriate classifier at a given point in time. The use of multiple classifiers, as described in this thesis, might be extended to deal with concept drift. Again, RDR may also be of use here to determine which are the most appropriate classifiers to use at a given time.

12.3.3 Real Time Classification

A real challenge for this work is to determine how best to automate this process for real time applications. For example, a farmer needs to know within days if

his crops are under threat, not after weeks of processing data. The techniques described in this work are ideal for applications such as agricultural mapping as they are fully automated.

Training times are still in the order of days rather than minutes or hours. Additional work on optimising the classification algorithms used should be investigated. As well, more detailed investigations of a given domain may be able to identify specific configurations and attributes that are more appropriate. By reducing the attribute search space classification time frames can most likely be reduced.

12.4 Conclusion

In this thesis we have demonstrated the use of a wide range of general classification techniques for extracting as much information as possible from the data available for training classification algorithms, and using only the most relevant information for the automatic generation and evaluation of accurate maps.

While this work focused on the classification of remotely sensed and ancillary data, the techniques described here are general enough to be applied to any domain, but are particularly relevant where training datasets are small, with noise or irrelevant attributes.

Specifically, this work demonstrated:

- Highlighting information in the available data.
- Neural network attribute selection and performance improvement.
- Simulating remotely sensed data.
- Combining a number of simple classifiers to improve classification accuracy.
- Automated generation and assessment of classifications of images.

Not only can we automatically generate reasonably reliable classifications for small, noisy datasets, we can also assess their relative accuracy with an acceptable degree of accuracy. In particular, the advantages of the techniques described in this work are:

- We can produce more consistent results between the test set and the entire image. That is, we can be reasonably confident that the error on the test set will be reflected in the error over the entire image.
- We can reduce the error in a classification by removing some of the misclassified pixels due to inconsistent classifications from a number of classifiers.
- Classifications can be generated and assessed automatically.

Bibliography

[1] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Mach. Learning*, 6:37–66, 1991.

[2] N. Ahuja, A. Rosenfeld, and R.M Haralick. Neighbor grey levels as features in pixel classification. *Pattern Recognition*, 12:251–260, 1980.

[3] K.M. Ali, P. Langley, M.A. Maloof, S. Sage, and T. Binford. Optimisation and simplification of hierarchical clusterings. In U.M. Fayyad and R. Utharusamy, editors, *Proc. 1998 Image Understanding Workshop*, pages 479–492, Monterey, CA, November 1998. Morgan Kaufmann.

[4] E.J. Van Allen, M.M. Menon, and P.N. Dicaprio. A modular architecture for object recognition using neural networks. In *Proc. Int. Neural Network conf. (INNC'90)*, pages 35–37, Paris, July 1990.

[5] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In R. Dale and M. Zock, editors, *Proc. 9th Nat. Conf. AI (AAAI'91)*, volume 2, pages 547–552, Anaheim, California, July 1991. AAAI Press / The MIT Press.

[6] H.I. Avi-Itzhak, J.A. Van Mieghem, and L. Rub. Multiple subclass pattern recognition: a maximum correlation approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(4):418–431, April 1995.

[7] J. Bala, K. de Jong, J. Huang, H. Vafaie, and H. Wechsler. Using learning to facilitate the evolution of features for recognizing visual concepts. *Special Issue of Evolutionary Computation: Evolution, Learning and Instinct: 100 Years of the Baldwin Effect*, 4(3):297–311, 1996.

[8] T. Baldwin, T. Tokunaga, and H. Tanaka. The parameter-based analysis of japanese relative clause constructions. *Information Processing Society of Japan SIG Notes*, 99(95):55–62, 1999.

[9] G.H. Ball and D.J. Hall. Isodata: A novel method of data analysis and pattern classification. Technical report, Stanford Research Institute, Menlo Par, CA, 1965.

[10] R. Battiti and A.M. Colla. Democracy in neural nets: voting schemes for classification. *Neural Networks*, 7(4):691–707, 1994.

[11] A. Baumgartner, C. Steger, C. Wiedemann, H. Mayer, W. Eckstein, and H. Ebner. Update of roads in GIS from aerial imagery: verification and multi-resolution extraction. *Int. Arch. Photogrammetry and Rem. Sens.*, XXXI(Part B3):53–58, 1996.

[12] H. Bischof, W. Schneider, and A.J. Pinz. Multi-spectral classification of landsat images using neural networks. *IEEE Transactions on Geosciences and Remote Sensing*, 30(3):482–490, May 1992.

[13] R.H. Blakeman. The identification of crop disease and stress by aerial photography. In M.D. Steven and J.A. Clark, editors, *Applications of Remote Sensing in Agriculture*, chapter 15, pages 229–254. Butterworths, 1990.

[14] A. Blum. *Neural networks in C++*. Wily-Interscience, New York, 1992.

[15] L. Bottou and P. Gallinari. A frameworkfor the co-operation of learning algorithms. In R. Lippman, J. Moody, and D.S. Touretzky, editors, *Neural Information Processing Systems 3 (NIPS-3)*, pages 781–788. Morgan Kaufmann, San Mateo, CA, 1991.

[16] L. Breiman and P. Spector. Submodel selection and evaluation in regression the x-random case. *Int. Stat. Rev.*, 60(3):291–319, 1992.

[17] T.M. Breuel. Recognition of handprinted digits using optimal bounded error matching. In *Proc. 2nd Int. Conf. Document Analysis and Recognition*, pages 493–496, Tsukwba, Japan, June 1993. IEEE.

[18] S.L. Caicco, J.M. Scott, B. Butterfield, and B. Csuti. A gap analysis of the management status of the vegetation of Idaho(U.S.A.). *Conservation Biology*, 9(3):498–511, 1995.

[19] R. Caruana. Algorithms and applications for multitask learning. In D. Fisher, editor, *Proc. 13th Int. Conf. on Machine Learning*, pages 87–95, Bari, Italy, 1996.

[20] R. Caruana and D. Freitag. Greedy attribute selection. In W.W. Cohen and H. Hirsh, editors, *Proc. 11th Machine Leaning Conf.*, pages 28–36, Rutgers University, New Brunswick, NJ, USA, July 1994. Morgan Kaufmann.

[21] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In U.M. Fayyad and R. Utharusamy, editors, *Proc. 1st Int. Conf. Knowledge Discovery and Data Mining*, pages 61–83, Montreal, Canada, August 1995. AAAI Press.

[22] K.S. Chen, Y.C. Tzeng, C.F. Chen, and W.L. Kao. Land-cover classification of mulitspectral imagery using a dynamic learning neural network. *Photogrammetric Engineering and Remote Sensing*, 61(4):403–408, 1995.

[23] Q. Chen. *Image Processing and Machine Learning for Histopathological Diagnosis.* PhD thesis, University of New South Wales, School of Computer Science and Engineering, Sydney, Australia, 1997.

[24] Q. Chen, A. Taylor, and J. Yong. Exploring renal biopsy sections: recognition of tubules and cell nuclei. In Yongnum Kim, editor, *Proc. SPIE Med. Img.*, Newport Beach, CA, USA, February 1997. SPIE.

[25] S.W. Cheon and S.H. Chang. Application of connectionist expert system for transient identification in nuclear power plants. In D. De Schreye, editor, *Proc. Pacific Rim Int. Conf. A.I. (PRICAI-92)*, pages 624–630, Seoul, Korea, 1992. Morgan Kaufmann.

[26] C.B. Chittineni. Learning with imperfectly labelled patterns. *Pattern Recognition*, 12:281–291, 1980.

[27] M.H. Coen. Building brains for rooms: Desigining distributed sofware agents. In *Proc. 9th Conf. Innovative Applications of AI (IAAI'97)*, pages 971–977, Providence, RI, 1997. AAAI Press / MIT Press.

[28] R.N. Colwell, editor. *The Manual of Remote Sensing.* Sheridan Press, Falls Church, Virginia, 1983.

[29] P. Compton, G. Edwards, B. Kang, L. Lazarus, R. Malor, T. Menzies, and P. Preston. Ripple down rules: Possibilities and limitations. In J.H. Boose and B.R. Gaines, editors, *Proc. 6th AAAI Knowledge Acquisition for Knowledge Based Systems Workshop*, pages 6.1–6.18, Bannf, 1991.

[30] P. Compton and R. Jansen. Knowledge in context: A strategy for expert system maintenance. In C.J. Barter and M.J. Brooks, editors, *Proc. 2nd Australian Joint Artificial Intelligence Conference*, pages 283–297, Adelaide, Australia, 1988. Springer-Verlag.

[31] P. Compton and R. Jansen. A philosophical basis for knowledge acquisition. *Knowledge Acquisition*, 2:241–257, 1990.

[32] C. Conese and F. Maselli. Selection of optimum bands from TM scenes through mutual information analysis. *J. Photogram. Rem. Sens.*, 48(3):2–11, 1993.

[33] R.G. Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sens. Environ.*, 37:35–46, 1991.

[34] C. Cortes, H. Drucker, D. Hoover, and V. Vapnik. Capacity and complexity control in predicting the spread between borrowing and lending interest rates. In U.M. Fayyad and R. Utharusamy, editors, *Proc. 1st Int. Conf. Knowledge Discovery and Data Mining*, pages 51–56, Montreal, Canada, August 1995. AAAI Press.

[35] T.M. Cover and P.E. Hart. Nearest neighbour pattern classification. *Trans. Information Th.*, 13:21–27, 1967.

[36] M.W. Craven and J.W. Shavlik. Learning symbolic rules using artificial neural networks. In *Proc. 10th Int. Conf. Machine Learning*, pages 73–80, San Mateo, CA, 1993. Morgan Kaufmann.

[37] M.W. Craven and J.W. Shavlik. Using sampling and queries to extract rules from trained neural networks. In W.W. Cohen and H. Hirsh, editors, *Proc. 11th Machine Learning Conf.*, pages 37–45, Rutgers University, New Brunswick, NJ, USA, July 1994. Morgan Kaufmann.

[38] R.E. Crippen. Calculating the vegetation index faster. *Remote Sens. Environ*, 34:71–73, 1990.

[39] E.P. Crist, R. Laurin, and R.C. Cicone. Vegetation and soils information contained in transformed thematic mapper data. In K.I. Itten, editor, *Proc. IGARSS*, pages 1465–1470, Zurich, Swtizerland, Sept 1986.

[40] J. Desachy, E.H. Zahzah, M. Zehana, and S. Castan. Production rules and neural networks for satellite image interpretation. In T. Ishida, editor, *Proc.*

Pacific Rim Int Conf. on AI (PRICAI'92), pages 1132–1138, Seoul Korea, 1992.

[41] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization. *Machine Learning*, 40(2):139–157, 2000.

[42] H. Drucker and C. Cortes. Boosting and other machine learning algorithms. In W.W. Cohen and H. Hirsh, editors, *Proc. 11th Machine Learning Conf.*, pages 53–61, Rutgers University, New Brunswick, NJ, USA, July 1994. Morgan Kaufmann.

[43] H. Drucker, R. Schapire, and P. Simard. Improving performance in neural networks using a boosting algorithm. In S. Hanson, J. Cowan, and L. Giles, editors, *Advances in Neural Information Processing 5 (NIPS-5)*, pages 42–49. Morgan-Kaufmann, 1993.

[44] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. Wiley, New York, 1973.

[45] B. Efron. Estimating the error rate of prediction rule: improvement on cross-validation. *J. Am. Stat. Ass.*, 78(382):316–330, 1983.

[46] C.D. Elvidge and Z. Chen. Comparison of broad-band and narrow-band red and near-infrared vegetation indices. *Remote Sens. Environ.*, 54:38–48, 1995.

[47] C.D. Elvidge and R.J.P. Lyon. Influence of rock–soil spectral variation on the assessment of green biomass. *Remote Sens. Environ.*, 17:265–279, 1985.

[48] B.K. Ersbell. A simple contextual classifier. In *8th Aust. Rem. Sens. Conf.*, pages 169–176, 1996.

[49] J.E. Estes and D.W. Mooneyhan. Of maps and myths. *Photogrammetric Engineering and Remote Sensing*, 60(5):517–524, 1994.

[50] F.H. Evans, P. Caccetta, and R. Ferdowsian. Integrating remotely sensed data with other spatial data sets to predict areas as risk from salinity. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 18–25, 1996.

[51] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Urthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. MIT Press/AAAI Press, 1996.

[52] U.M. Fayyad. Data mining and knowledge discovery in databases: applications in astronomy and planetary science. In *Proc. 13th Conf. A.I.*, pages 61–83, 1996.

[53] D. Fisher. Optimisation and simplification of hierarchical clusterings. In U.M. Fayyad and R. Utharusamy, editors, *Proc. 1st Int. Conf. Knowledge Discovery and Data Mining*, pages 118–123, Montreal, Canada, August 1995. AAAI Press.

[54] R.W. Fitzgerald and B.G. Lees. Assessing the classification accuracy of multi-source remote sensing data. *Remote Sens. Environ.*, 47:362–368, 1994.

[55] P. Flemons. Monitoring vegetation change in NSW : A comparison of change detection techniques. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 270–278, 1996.

[56] G.M. Foody, M.B. McCulloch, and W.B. Yates. Classification of remotely sensed data by an artificial neural network: issues related to training data characterisitics. *Photogrammetirc Engineering and Remote Sensing*, 61(4):391–401, 1995.

[57] Australia Forestry Commission of New South Wales, Sydney. Forest types in new south wales, research note no. 17. Technical report, Forestry Commission of New South Wales, 1989.

[58] P.G. Foschi. A geometric approach to a mixed pixel problem : Detecting subpixel woody vegetation. *Remote Sens. Environ.*, 50:317–327, 1994.

[59] L. Fox, J.A. Brockhaus, and N.D. Tosta. Classification of timberland productivity in north western California using Landsat, topographic and ecological data. *Photogrammetric Engineering and Remote Sensing*, 51(11):1745–1752, 1985.

[60] K. Fukunaga. *Statistical Pattern Recognition.* Academic Press, San Diego, 1990.

[61] M. Gahegan, G. German, and G. West. Improving neural network performances on the classification of complex geographic datasets. *Geographical Systems*, 1(1):3–22, 1999.

[62] G.D. Garson. Interpreting neural-network connection weights. *AI Expert*, 6(4):47–51, April 1991.

[63] G.W.H. German and M.N. Gahegan. Neural network architectures for the classification of temporal image sequences. *Computers and Geosciences*, 22(9):969–979, November 1996.

[64] J. Goebel, K. Volk, H. Walker, F. Gerbault, P. Cheeseman, M. Self, J. Stutz, and W. Taylor. A Bayesian classification of the IRAS LRS atlas. *Astronomy and Astrophysics*, 222:L5–L8, 1989.

[65] D.G. Goodenough, M. Goldberg, G. Plunkett, and J. Zelek. An expert system for remote sensing. *IEEE Trans. Geosci. Rem. Sens.*, GE-25(3):349–359, 1987.

[66] H.K. Greenspan and R. Goodman. Remote sensing image analysis via a texture classification neural network. In S. Hanson, J. Cowan, and L. Giles, editors, *Advances in Neural Information Processing 5 (NIPS-5)*, pages 425–432. Morgan-Kaufmann, 1993.

[67] G. Grigg, H. McCallum, A. Taylor, and G. Watson. Monitoring frog communities: an application of machine learning. In W. Wahlster, editor, *Proc. 8th Innovative App. AI Conf.*, pages 1564–1569, Portland, Oregon, USA, August 1996.

[68] H. Guo and S.B. Gelfand. Classification trees with neural network feature selection. *IEEE Trans. Neural Networks*, 3(6):923–933, 1992.

[69] R. Hanson, J. Stutz, and P. Cheeseman. Bayesian classification theory. Technical Report FIA-90-12-7-01, Artificial Intelligence Branch, NASA Ames Research Centre, Moffet Field, CA, December 1994.

[70] H. He and Z. Huang. Boosting neural networks in real world applications: an empirical study. In A. Satter, editor, *Proc. 10th Aust. Joint Conf. AI*, pages 321–329, Armidale, Australia, 1997. World Scientific Publishing.

[71] C.M. Higgins and R.M. Goodman. Learning fuzzy rule-based neural networks for control. In S. Hanson, J. Cowan, and L. Giles, editors, *Advances in Neural Information Processing 5 (NIPS-5)*, pages 350–357. Morgan-Kaufmann, 1993.

[72] A.R. Huete, R.D. Jackson, and D.F. Post. Spectral response of a plant canopy with different soil backgrounds. *Remote Sens. Environ.*, 17:37–53, 1985.

[73] C.A. Iglesias, J.C. Gonzalez, and J.R. Velasco. A fuzzy-neural multiagent system for optimization of a roll-mill application. In *Proc. 11th Int. Conf. Industrial and Engineering Applications of AI and Expert Systems*, pages 596–605. LNCS vol 1415, Springer Verlag, 1996.

[74] F. Ince. Maximum likelihood classification, optimal or problematic? a comparison with the nearest neighbour classification. *Int. J. Rem. Sens.*, 8(12):1829–1838, 1987.

[75] C.H. Jarvis and N. Stuart. The sensitivity fo a neural network for classifying remotely sensed imagery. *Computers and Geosciences*, 22(9):959–967, November 1996.

[76] X. Jia and J.A. Richards. Feature reduction using a supervised hierarchical classifier. In *Proc. 8th Aust. Rem. Sens. Conf.*, 1996.

[77] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In W.W. Cohen and H. Hirsh, editors, *Proc. 11th Machine Learning Conf.*, pages 121–129, Rutgers University, New Brunswick, NJ, USA, July 1994. Morgan Kaufmann.

[78] T. Kailath. The divergence and Bhattacharyy distance measures in signal selection. *IEEE Trans. Communication Theory*, COM-15:52–60, 1967.

[79] M. Kaiser and J. Kreuziger. Integration of symbolic and connectionist learning to ease robot programming and control. In *EACI'94 Workshop on Combining Symbolic and Connectionist Processing*, pages 20–29, Amsterdam, Netherlands, August 1994. Harvard University Press.

[80] B. Kanefsky, J. Stutz, and P. Cheeseman. An improved automatic classification of a Landsat/TM image from Kansas (FIFE). Technical Report FIA-94-01, Artificial Intelligence Branch, NASA Ames Research Center, Moffet Field, CA, 1994.

[81] P.M. Kelly and J.M White. Preprocessing remotely-sensed data for efficient analysis and classification. In U.M. Fayyad and R. Uthurusamy, editors, *Applications of AI : Knowledge-based systems in Aerospace and Industry*, pages 24–30, Orlando, Florida, April 1993. SPIE.

[82] K. Kira and L.A. Rendell. A practical approach to feature selection. In D. Sleeman and J.E. Edwards, editors, *Proc. 9th Machine Leaning Conf.*, pages 249–256, Aberdeen, Scotland, 1992. Morgan Kaufmann.

[83] M. Klusch. HNS - a hybrid neural system and its use for the classification of stars. In *Proc. 1993 Int. Joint Conf. on Neural Networks (IJCNN'93)*, pages 687–692, Nagoya, Japan, 1993. IEEE.

[84] D.E. Knuth. *The Art of Computer Programming*, volume 2. Addison Wesley, Reading, Massachusetts, 1969.

[85] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In N. Lavrac and S. Wrobel, editors, *Proc. 14th Int. Joint Conf. AI (IJCAI'95)*, pages 1137–1143, Montreal, Canada, Aug 1995. Morgan Kaufmann.

[86] D. Lamb. Airborne video and spatial data. In *Proc. 25th Riverina Outlook Conf.*, pages 45–54, Wagga Wagga, Australia, 1995.

[87] D.W. Lamb, J. Louis, and G. McKenzie. The development and application of an airborne video system for resource management. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 29–34, 1996.

[88] J.R. Landis and G.C. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174, 1977.

[89] P. Langley. Selection of relevant features in machine learning. In J.C. Boudreaux, B.W. Hamill, and R.N. Jernigan, editors, *Proc. AAAI Fall Symposium on Relevance*, pages 379–382, New Orleans, 1994. AAAI Press.

[90] P. Langley and S. Sage. Induction of selective Bayesian classifiers. In P. Besnard and S. Hanks, editors, *Proc. 10th Conf. Uncertainty in AI*, pages 399–406, Seattle, WA, 1994. Morgan Kaufmann.

[91] V.J. LeBau and H.T. Schreuder. A multi-phase, multiresource inventory procedure for assessing renewable natural resources and monitoring change. In H. Bassirirad, editor, *Renewable Resource Inventories for Monitoring Changes and Trens, Proc. Soc. Amer. Foresters*, pages 456–459, Anchorage, Alaska, 1993.

[92] B.G. Lees. Neural network applications in the geosciences: an introduction. *Computers and Geosciences*, 22(9):955–957, November 1996.

[93] D. Legitimus and L. Schwab. Natural underwater sounds identification by the use of neural networks and linear techniques. In H. Abut, editor, *Proc. Int. Neural Network Conf.*, pages 123–126, New York, 1990. IEEE Press.

[94] S. Lehar, T. Howells, and I. Smotroff. Application of grossberg and mingolla neural vision model to satellite weather imagery. In H. Abut, editor, *Proc. Int. Neural Network Conf.*, pages 805–808, New York, 1990. IEEE Press.

[95] T.M. Lillesand and R.W. Keifer. *Remote Sensing and Image Interpretation.* Wiley & Sons, New York, 1994.

[96] R.P. Lippmann. An introduction to computing with neural networks. *Computer Architecture News ACM*, 16(1):7–25, 1988.

[97] W. Malina. Some multi-class Fisher feature selection algorithms and their comparison with karhunen-loeve algorithms. *Pattern Recognition Letters*, 6(5):279–285, 1987.

[98] M.A. Maloof, P. Langley, T. Binford, and S. Sage. Improving rooftop detection in aerial images through machine learning. Technical report, Institute for the Study of Learning and Expertise, Palo Alto, CA, 1998.

[99] T.J. Malthus, B. Andrieu, F.M. Danson, K.W. Jaggard, and M.D. Steven. Candidate high spectral resolution infrared indices for crop cover. *Remote Sens. Environ.*, 46:204–212, 1993.

[100] S.E. Marsh, J.L. Walsh, and C. Sobrevila. Evaluation of airborne video data for land-cover classification accuracy assessment in an isolated Brazilian forest. *Remote Sens. Environ.*, 48:61–69, 1994.

[101] P. Masson and W. Pieczynski. SEM algorithm and unsupervised statisitcal segmentaiton of satellite images. *IEEE Trans. Geoscience and Rem. Sens.*, 31(3):618–633, 1993.

[102] S. Mathieu-Marni, P. Leymarie, and M. Berthod. Removing ambiguities in a multi-spectral image classification. *Int. J. Rem. Sens.*, 17(8):1493–1504, 1996.

[103] I. McGowen, G. Dudgeon, I. Smith, S. McGrath, and F. Brandsema. Remote sensing and GIS for better land management in the Jemalong irrigation district and surrounding areas. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 230–236, 1996.

[104] D.M. McKeown, G.E. Bulwinkle, S.D. Cochran, S.J. Ford, S.J. Gifford, W.A. Harvey ad Y.C. Hsieh, C. McGlone, W.A. Harvey, M.F. Polis, M. Roux, and J.A. Shufelt. Research in the automated analysis of remotely

sensed imagery: 1994-1995. In R.S. Engelmore and A.J. Morgan, editors, *Proc. ARPA Image Understanding Wkshp*, Palm Springs, CA, 1996. Morgan Kaufmann.

[105] R.A. Mead and J. Szajgin. Landsat classification accuracy assessment procedures. *Photogrammetric Engineering and Remote Sensing*, 48(1):139–141, 1982.

[106] R.S. Michalski. Plausible justification trees: A framework for deep and dynamic integration of learning strategies. *Mach. Learn.*, 11:237–261, 1993.

[107] D. Michie. Current developments in expert systems. In R.J. Scherer, editor, *Proc. 2nd Australian Conference on Applications of Expert Systems*, pages 163–182, Sydney, Australia, May 1986.

[108] D. Michie, D.J. Spiegelhalter, and C.C Taylor. *Machine learning, neural and statistical classification.* Ellis Horwood, 1994.

[109] J.R. Milford and G. Dugdale. Estimation of rainfall using geostationary satellite data. In M.D. Steven and J.A. Clark, editors, *Applications of Remote Sensing in Agriculture*, chapter 5, pages 97–110. Butterworths, 1990.

[110] L.K. Milne. Feature selection using neural networks with contribution measures. In Xin Xao, editor, *AI'95 Poster Proceedings*, University College, UNSW ADFA, Canberra, November 1995.

[111] L.K. Milne. Attribute selection in neural networks used to classify remotely sensed data. In H. Yan, D.D. Feng, and J. Jin, editors, *Proc. Visual Information Processing Wkshp*, pages 21–26, Sydney, Australia, December 1997. University of Sydney.

[112] L.K. Milne. Improving classification accuracy of machine learning techniques applied to remotely sensed data. In P. Collier, editor, *Proc. Applications Track 11th Aust. Joint Conf. AI (AI'98)*, pages 26–37, Brisbane, Australia, July 1998. Griffith University.

[113] L.K. Milne, T.D. Gedeon, and A.K. Skidmore. Classifying dry sclerophyll forest from augmented sattelite data : Comparing neural network, decision tree and maximum likelihood. In M. Charles and C. Latimer, editors, *Proc. 6th Australian Conference on Neural Networks (ACNN'95)*, pages 160–163, Sydney, Australia, February 1995. Dept. Electrical Engineering, University of Sydney.

[114] R. Molina, N. Perez de la Blanca, and C.C. Taylor. Modern statistical techniques. In D. Michie, D.J. Spiegelhalter, and C.C Taylor, editors, *Machine learning, neural and statistical classification.*, chapter 4, pages 29–49. Ellis Horwood, 1994.

[115] J.M.J. Murre. *Learning and Categorization in Modular Neural Networks.* Harvester Wheatsheaf, Hillsdale, NJ, 1992.

[116] C.M.U. Neale and B.G. Crowther. An airborne multi-spectral video/radiometer remote sensing system: Development and calibration. *Remote Sens. Environ.*, 49:187–194, 1994.

[117] E.M. Nel, C.A. Wessman, and T.T. Veblen. Digital and visual analysis of thematic mapper imagery for differencing old growth from younger spruce-fir stands. *Remote Sens. Environ.*, 48:291–301, 1994.

[118] G.J.A. Nieuwenhuis, J.W. Miltenburg, and H.A.M. Thunnissen. Application of remote sensing and geographical information systems in water management. In M.D. Steven and J.A. Clark, editors, *Applications of Remote Sensing in Agriculture*, chapter 6, pages 111–123. Butterworths, 1990.

[119] C. Nikolopoulos and P. Fellrath. Hybrid expert system for investment advising. *Expert Systems*, 11(4):245–248, 1994.

[120] T.W. Norton, B.G. Mackey, and D.B. Lindenmayer. Comments on biological and environmental data sets required for the australian nation wilderness inventory. *Australian Forestry*, 53(2):124–130, 1990.

[121] S.J. Nowlan and G.E. Hinton. Evaluation of adaptive mixtures of competing experts. In R. Lippman, J. Moody, and D.S. Touretzky, editors, *Neural Information Processing Systems 3 (NIPS-3)*, pages 774–781. Morgan Kaufmann, San Mateo, CA, 1991.

[122] A.L. O'Neill. Vegetation indices for different seasonal conditions in semi–arid Australia. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 143–151, 1996.

[123] I. Overton. Detecting vegetation change on a semi-arid floodplain using Landsat TM and MSS, Chowilla, South Australia. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 78–85, 1996.

[124] R. Pearson, J. Grace, and G. May. Real-time airborne agricultural monitoring. *Remote Sens. Environ.*, 49:304–310, 1994.

[125] T.L. Phillips(*ed.*). LARSYS version 3 users manual. Laboratory for applications of remote sensing. Technical report, Purdue University, West Lafayette, IN, 1973.

[126] Piatetsky-Shapiro and W. Frawley. *Knowledge Discovery in Databases*. MIT Press, 1991.

[127] B. Pinty, C. Leprieur, and M.M. Verstraete. Towards a quantitative interpretation of vegetation indices. Part 1: Biophysical canopy properties and classical indices. *Rem. Sens. Rev.*, 7:127–150, 1993.

[128] B. Pinty and M.M. Verstraete. Gemi : a non-linear index to monitor global vegetation from satellites. *Vegetatio*, 101:15–20, 1992.

[129] J.R. Potter, D.K. Mellinger, and C.W. Clark. Marine mammal call discrimination using artificial neural networks. *J. Accoust. Soc. Am.*, 96(3):1255–1262, 1994.

[130] J.C. Price. How unique are spectral signatures? *Remote Sens. Environ.*, 49:181–186, 1994.

[131] J. Qi, A.R. Huete, Y.H. Kerr, and S. Sorooshian. A modified soil adjusted vegetation index. *Remote Sens. Environ.*, 48:119–126, 1994.

[132] J.R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan-Kaufmann, San Mateo, CA, 1993.

[133] R.B. Rao, T.B. Voight, and T.W. Fermanian. Data mining of subjective agricultural data. In *Proc. 10th Int. Conf. Machine Learning*, pages 244–251, 1993.

[134] J.A. Richards. *Remote Sensing Digital Image Analysis : An Introduction*. Springer–Verlag, New York, 1994.

[135] W.J. Ripple. Determining coniferous forest cover and forest fragmentation with NOAA-9 advanced very high resolution radiometer data. *Photogrammetric Engineering and Remote Sensing*, 60(5):533–540, 1994.

[136] G. Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

[137] R. Rohwer, M. Wynn-Jones, and F. Wysotzki. Neural networks. In D. Michie, D.J. Spiegelhalter, and C.C Taylor, editors, *Machine learning, neural and statistical classification.*, chapter 6, pages 84–106. Ellis Horwood, 1994.

[138] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing. Volume 1: Foundations.* MIT Press, Cambridge, MA, 1986.

[139] C. Sammut. Knowledge representation.. In D. Michie, D.J. Spiegelhalter, and C.C Taylor, editors, *Machine learning, neural and statistical classification.*, pages 228–245. Ellis Horwood, 1994.

[140] H.T. Schreuder, V.J. LaBau, and J.W. Hazard. The Alaska four-phase forest inventory sampling design using remote sensing and ground sampling. *Photogrammetric Engineering and Remote Sensing*, 61(3):291–297, 1995.

[141] J.W. Shavlik and G.G. Towell. An approach to combining explanation based and neural learning algorithms. *Connection Sci.*, 1(3):231–253, 1989.

[142] V.K. Shettigara. Semi-automatic detection of changes due to artificial objects in multi-spectral images. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 159–168, 1996.

[143] J. Sietsma and R.J.F. Dow. Creating artificial neural networks that generalize. *Neural Networks*, 4:67–79, 1991.

[144] V.G. Sigillito and L.V. Hutton. Case study II: radar signal processing. In R.C. Eberhart and R.W. Dobbins, editors, *Neural network PC tools*, pages 235–250. Academic Press, 1990.

[145] A.K. Skidmore. Unsupervised training area selection in forests using non-parametric distance measure and spatial information. *Int. J. Rem. Sens.*, 10(1):147–169, 1989.

[146] A.K. Skidmore, W. Brinkhof, J. Delaney, and B.J. Turner. Using neural networks to analyse spatial data. In *Proc. 7th Aust. Rem. Sens. Conf.*, pages 235–246, 1994.

[147] A.K. Skidmore, E. Knowles, and B.J. Turner. Neural networks for image processing. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 169–175, 1996.

[148] A.K. Skidmore and B.J Turner. Forest mapping accuracies are improved using a supervised nonparametric classifier with SPOT data. *Photogrammetric Engineering and Remote Sensing*, 54(10):1415–1421, 1988.

[149] A.K. Skidmore, G.B. Wood, and K.R. Shepherd. Remotely sensed digital data in forestry: a review. *Aust. Forestry*, 50(1):40–53, 1986.

[150] R.F. Smith and A.L. O'Neill. Discrimination of pasture weeds using spectral analysis. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 58–62, 1996.

[151] R.O. Squire. The professional challenge of balancing sustained wood production and ecosystem conservation in the native forests of south-eastern Australia. *Aust. For.*, 56(3):237–248, 1993.

[152] C. Steger, W. Eckstein, and C. Weidemann. Update of roads in GIS by automatic extraction from aerial imagery. In *Proceedings of the Second International Airborne Remote Sensing Conference and Exhibition*, volume III, pages 308–317, San Francisco, June 1996. Environmental Research Institute of Michigan.

[153] M.D. Steven. Satellite remote sensing for agricultural management: opportunities and logistic constraints. *ISPRS J. Photogrammetry and Remote Sensing*, 48(4):29–34, 1993.

[154] T.A. Stone, P. Schlesinger, R.A. Houghton, and G.M. Woodwell. A map of the vegetation of south america based on satellite imagery. *Photogrammetric Engineering and Remote Sensing*, 60(5):541–551, 1994.

[155] D.Z. Sui. Recent applications of neural networks for spatial data handling. *Canadian J. Rem. Sens.*, 20(4):368–380, 1994.

[156] P.H. Swain and S.M. Davis, editors. *Remote Sensing: The Quantitative Approach.* McGraw-Hill, Fonte, New York, 1978.

[157] R. Swann, D. Hawkins, A. Westwell-Roper, and W. Johnstone. The potential for automated mapping from geocoded digital image data. *Photogrammetric Engineering and Remote Sensing*, 54(2):187–193, 1988.

[158] S.S. Talbot and C.J. Markon. Vegetation mapping of Nowintra National Wildlife Refuge, Alaska using Landsat MSS digital data. *Photogrammetric Enginering and Remote Sensing*, 52(6):791–799, 1986.

[159] A. Taylor. Recognising biological sounds using machine learning. In A. Satter, editor, *Proc. 8th Aust. Joint Conf. AI*, Canberra, November 1995.

[160] G.D. Tecuci and R.S. Michalski. A method for multi-strategy task-adaptive learning based on plausible justifications. In D. Aha and P.J. Riddle, editors, *Proc. 8th Int. Wkshp. Machine Learning*, pages 549–553. Morgan Kaufmann, 1991.

[161] S. Thiria, C. Mejia, F. Badran, and M. Crepon. Multi-modular architecture for remote sensing operations. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing 4 (NIPS-4)*, pages 675–682. Morgan-Kaufmann, 1991.

[162] S.B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Dzeroski, S.E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Konoenko, J. Kreuziger, R.S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich, H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, and J. Zhang. The MONKS's problems: A performance comparison of different learning algorithms. Technical Report Comptuer Science Reports, CMU-CS-91-197, Carnegie Mellon University, Pittsburgh, PA, 1991.

[163] J.C. Trinder and H. Li. Semi-automatic feature extraction by snakes. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 194–201, 1996.

[164] C.J. Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sens. Environ.*, 8:127–150, 1979.

[165] J. Uebersax. Statistical methods for rater agreement. `http://ourworld.compuserve.com/homepages/jsuebersax/agree.htm`, 2005.

[166] E.J. van Allen. Application of boundary contour neural network to illusions and infrared sensor imagery. In *IEEE 1st Int. Conf. Neural Networks*, page 93, San Diego, 1987. IEEE Press.

[167] W. Wan and D. Fraser. A self-organising neural network framework for high dimensional data analysis. In *Proc. 7th Aust. Rem. Sens. Conf.*, pages 151–156, 1994.

[168] AutoClass WEBSITE. Bayesian learning group. nasa ames research center. the AutoClass project. `http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass`, 1999.

[169] GRASS WEBSITE. Geographic resources analysis support system. `http://www.cecer.army.mil/grass/GRASS.main.html`, 1999.

[170] UCI WEBSITE. UCI Machine Learning Repository. `http://www.ics.uci.edu/ mlearn/MLRepository.html`, 1999.

[171] G.M. Weiss and H. Hirsh. The problem with noise and small disjuncts. In J.W. Shavlik, editor, *Proc. 15th Int. Conf. Machine Learning*, pages 269–277, Madison, Wisconsin, 1998. Morgan Kaufmann.

[172] S.M. Weiss and N. Indurkhya. Decision tree pruning: biased or optimal. In B. Hayes-Roth and R. Korf, editors, *Proc. Nat. Conf. AI (AAAI'94)*, pages 626–632, Seattle, Washington, 1994. AAAI Press / MIT Press.

[173] S.M. Weiss and I. Kapouleas. An empirical comparison of pattern recognition, neural nets and machine learning classification methods. In *Proc. 9th Int. Joint Conf. AI (IJCAI'89)*, pages 781–787, San Francisco, 1989. Morgan Kaufmann.

[174] S.M. Weiss and C.A. Kulikowski. *Computer Systems that Learn.* Morgan Kaufmann, San Francisco, 1991.

[175] G.G. Wilkinson, S. Folving, I. Kanellopoulous, N. McCormick, K. Fullerton, and J. Mégier. Forest mapping from multi-source satellite data using neural network classifiers - an experiment in Portugal. *Rem. Sens. Rev.*, 12:83–106, 1995.

[176] P.A. Wilson. Local response neural networks for experimental classification of remotely sensed images. In *Proc. 8th Aust. Rem. Sens. Conf.*, pages 185–192, 1996.

[177] P.M. Wong, T.D. Gedeon, and I.J. Taggart. An improved technique in porosity prediction: A neural network approach. *IEEE Transactions on Geoscience and Remote Sensing*, (in press) 1995.

[178] X. Xu and Y. Yin. A neural network model for forest management. In *Proc. GIS Symp.*, pages 502–505, 1990.

[179] T. Yoshida and S. Omatu. Neural network approach to land cover mapping. *IEEE Trans. Geosci. Rem. Sens.*, 32(5):1103–1109, 1994.

[180] X. Yuan, D. King, and J. Vlcek. Sugar maple decline assessment based on spectral and textural analysis of multi-spectral aerial videography. *Remote Sens. Environ.*, 37:47–54, 1991.

Appendix A

Glossary

ABVS Airborne video system, developed at Charles Sturt University, Wagga Wagga.

Ancillary data Data used in addition to spectral data, that is, additional attributes.

attribute Any numeric or non-numeric value associated with a given pixel in an image. These may be the original spectral values, derived values or other ancillary information.

AVHRR Advanced very high resolution radiometer, a satellite sensing platform.

AVRIS Airborne visible/infrared imaging spectrometer.

band/spectral band A remotely sensed image, as used here, is of an area on the ground with each pixel having a number of reflectance values measured. Each reflectance value is in a given range of the electromagnetic spectrum and is referred to as a spectral band.

binary classifier A classifier that gives a `yes` or `no` classification.

broad band Reflectance values in a given band are measured over a given range of the electromagnetic spectrum, and when the range of values is large the measurement is referred to as broad band.

case/training case Corresponds to a pixel in an image. Each pixel has a number of attributes associated with it and a class label.

class/class label A category assigned to a given pixel, either by a classifier system or human interpretation.

classification A broad term encompassing the actual classification

CSU The Charles Sturt university dataset.

default classification A trained classifier that has not been able to separate the characteristics of the classes in the training data. Each input case is given a default class membership regardless of its actual attribute values.

DEM Digital elevation model, see elevation model.

elevation model An elevation model is a representation of the height and shape of the Earth's surface.

epoch When training a neural network one full presentation of the training data to the neural network is referred to as a epoch.

ground truth Ground truth data is essentially the collection of training data from ground surveys of the study area. Knowing what is on the ground at a given point gives us a class label that can be combined with a set of attributes and used as training data. The type and quality of the data is entirely dependant on the experience of the team carrying out the survey, and may vary between teams. This is added to the fact that reasonably cheap and accurate positioning systems (GPS) have only recently become available and in the past the actual positions of these sites were estimated using maps.

IBL Instance-based learning.

Landsat TM Landsat thematic mapper, a satellite sensing platform.

Landsat MSS Landsat multi-spectral scanner, a satellite sensing platform.

map A classification of a given area that may have been generated manually by a human expert, in combination with ground truth surveys or even historical data, or a computer generated classification.

NIR Near-infrared.

NN Neural network.

NSW New South Wales.

orthoimage An orthoimage is generated by rectifying distortions in an image caused by variations in the height of the terrain. It contains pixels that are all to the same scale.

overfitted/overgeneralised The classifier has learnt the details of the training data and has not found a generalisation of the training data.

remotely sensed image Data from any remote sensing platform with any number of spectral bands.

resolution The resolution data refers to the area on the ground that each pixel corresponds to. Low resolution data represents a large area for each pixel and high resolution data represents a small area.

RNP The Royal National Park dataset. This is for a small area within the park, called Audley.

SIM The simulated remotely sensed image dataset.

single class classifier A classifier that gives an `in` or `notin` classification for a given class. The classifier is trained on data with two classes - the `in` class which contains training cases for a single class and the `notin` class which contains training cases from all other known classes. The aim is to train a classifier that will recognise the characteristics of the most consistent class.

specialisation The classifier has learnt the details of the training data and has not found a generalisation of the training data.

SPOT System Probatoire d'Observation de la Terre, a French satellite sensing platform.

stop set Used only in the training of the neural network. The specific set of labelled cases that are used to determine the stopping point in training. They are presented to the classifier after training and the point at which the error on the stop set starts increasing is the point at which to stop training.

test set The specific set of labelled cases that are not used in at all in training a classifier. They are, however, presented to the classifier after training has been completed to estimate the error of the classifier on data for which the class is not known.

topographic maps Topographic maps contain natural features such as hills and rivers, as well as cultural features such as roads, bridges and railways.

training data The entire labelled data set, split into training, stop and test sets.

training set The specific set of labelled cases that are presented to a classifier for training.

two class classifier A classifier that has been trained on data containing only two classes. This attempts to distinguish between two classes.

UCI University of California, Irvine.

unseen data Refers to data that is not used at all during the training of a classifier, but rather is presented to the trained classifier for classification.

Appendix B

Publications

Publications from this work can be found via the following links. The work described in these papers has been incorporated into the thesis in the relevant sections.

L.K. Milne, T.D. Gedeon, and A.K. Skidmore. Classifying dry sclerophyll forest from augmented satellite data : Comparing neural network, decision tree and maximum likelihood. In *Proc. 6th Australian Conference on Neural Networks*, Sydney, pages 160–163, February 1995.
`http://handle.unsw.edu.au/1959.4/37616`

L.K. Milne. Feature selection using neural networks with contribution measures. In *AI'95 Poster Proceedings*, Canberra, November 1995.
`http://handle.unsw.edu.au/1959.4/37628`

L.K. Milne and C. Willock. Comparison of two methods for increasing training set size for neural networks. In *AI in the Environment Wkshp*, Canberra, pages 89–94, November 1995.
`http://handle.unsw.edu.au/1959.4/37622`

L.K. Milne. Attribute selection in neural networks used to classify remotely sensed data. In *Visual Information Processing Wkshp*, Sydney, pages 21–26, December 1997.
`http://handle.unsw.edu.au/1959.4/37610`

L.K. Milne. Improving Classification Accuracy of Machine Learning Techniques applied to Remotely Sensed Data. In *Proc AI'98*, Brisbane, pages 26–37, July 1998.
http://handle.unsw.edu.au/1959.4/37655

Appendix C

Recent Developments

A review of the research published since the submission of this thesis is given here.

C.1 Multi-Strategy Classification Schemes

Multi-strategy classification schemes, such as that given in this thesis, continue to be discussed in the machine learning literature.

Liu et al. [LCJM04] describe a classifier scheme using combinations of neural networks with attribute selection that has many similarities with the work discussed in this thesis.

The classification scheme starts by pre-processing the data. The data is then resampled 100 times using bootstrapping – a new training set is created by selecting a subset of cases with replacement from the original training set, this new training set is then used to train a new classifier [Die00]. The classifier used was three co-operative and competitive neural networks. Attribute selection is first carried out using a ranksum test, principle components analysis and a t-test to determine the most relevant attributes and give three different datasets to train the three different neural networks. The average output of the three networks is the output given.

The resampling is carried out for 100 iterations to give 100 subsets of the data to train the co-operative and competitive neural networks, which gives 100 classifications for each instance presented. The final classification given to a single

instance is simply a majority vote of the 100 classifications given.

Three publicly available medical datasets were used to test the classification scheme. As with this thesis, each of the classification problems were reduced to binary classifications.

The results of the classification scheme were compared against other published results for the given datasets, as well as compared against bagged decision trees [Die00]. In all cases the multi-strategy classification scheme used by Liu et al. was an improvement over other techniques described. Comparison with leave one out cross validation on the datasets gave the similar error rates to the multi-strategy classification scheme and so provided support for the validity of the improvements achieved.

Liu et al. concluded that the improved error rates were largely due to the attribute selection. One of the aims in this thesis was to create different views of the data to enable better classification. The method of attribute selection used by Liu et al. is an alternative way of achieving this. They also state that the advantage of using more than one classifier with a majority voting scheme is that it reduces the effects of noise in the data and identifies the central features of each class, which is in complete agreement with the work done in this thesis.

In contrast to the work discussed in this thesis, they used a combination of 100 neural network classifiers. While the results are clearly an improvement over other classification schemes the large number of classifiers being trained and used might result in long execution times, although this is not discussed. Given that the datasets were all less than 1,000 instances in total, training and classification times are not likely to be an issue. However, in a remote sensing or other imagery based domains, where it is not uncommon to have hundreds of thousands of instances, the efficiency of this classification scheme given might be prohibitive.

Lee and Ersoy [LE07] also discuss a classification scheme that used multiple neural networks, each trained by varying the data or other training parameters. The outputs of each trained neural network were combined by using a weighted sum to generate a single output. Similar to the results obtained in this thesis classification accuracy improved over using a single classifier. They state that the improvement of classification performance by consensus is achieved when the errors produced by multiple classifiers are different, and there is little correlation between them. While the classification scheme of Lee and Ersoy was different to that given in

this thesis their findings and conclusions are in complete agreement with those reported in this thesis.

At the time of writing the author was unable to find work that was directly comparable to the work carried out in this thesis in remote sensing or vegetation mapping domains, and this continues to be the case. Hybrid classification schemes are used in these domains, but nothing could be found that is similar to the framework introduced in this work. The focus of hybrid approaches tends to be more on statistically based methods or methods for pre-processing the data before using a single classifier. A sample of the publications found discussing such hybrid techniques include [KKOL00, SMS00, DWBS$^+$04, LVK$^+$05, ACN07, CCP07].

Neural networks have been popular in remote sensing domains for some time, and while not widespread there is evidence that other machine learning techniques and multi-strategy classification schemes are beginning to be used.

Boosting and bagging [Die00] have become more commonly used techniques which are also being discussed within the remote sensing literature. These techniques have become popular due to the reported improvements in classification accuracy. In particular they are of interest in remote sensing applications because they overcome some of the drawbacks in using more traditional statistical approaches [GBS06].

Bagging is based on training many classifiers on bootstrapped samples from a training set (such as that discussed in [LCJM04]). Bagging has been shown to be effective on datasets for which the classifier is unstable, that is, small changes in the training set cause large changes in the classification results. The small, noisy training sets discussed in this thesis fall into this category, particularly when used with neural networks. Bagging has been demonstrated to reduce the variance of the classification and reduce the error rates [GBS06].

In this thesis, the use of multiple classifiers with attribute selection has been used to serve the same result as bagging. In the case of neural networks the use of binary classifiers in combination with contribution analysis increased the stability of the trained classifiers. In addition, the use of contribution analysis requires only two iterations of training, rather than a large number of iterations as is required with bagging.

Boosting uses iterative re-training of a classifier, where the incorrectly classified samples are given increased weighting as the iterations progress. Boosting pro-

duces more accurate classifications but is slow, tends to overfit the data and is sensitive to noise [Die00, GBS06]. This makes it an unsuitable approach for the types of data discussed in this thesis. This conclusion seems to be supported by the remote sensing literature as bagging was referred to more often.

Breiman [Bre01] introduced a technique called random forests based on bagging and the random subspace method of Ho [Ho98]. The random forest algorithm creates multiple decision trees, each trained on a bootstrapped sample of the original training data. It searches a randomly selected subset of the input attributes to determine a split for each node in the decision tree. A majority vote of each of the classifiers trained in this way is used to give the final classification for a given input.

There does continue to be some evidence that statistically based techniques are not as effective in classifying remotely sensed data, particularly when non-numerical data is being used as well. Gislason et al. [GBS06] used random forests to classify remotely sensed data and reported improved classification accuracies over more traditional statistical techniques. Lower error rates were obtained when using boosting and bagging however the faster training times of random forests were considered a reasonable trade off for the relatively small increases in the error rates when using random forests. The random forest algorithm is also preferred as it can also be used to determine mislabelled data.

Gislason et al. recommend the use of random forests for classification of multisource remotely sensed data, particularly where appropriate statistical models cannot be used. However, random trees are a less general approach than that discussed in this thesis which is not limited to decision tree classifiers. The common ground with the work in this thesis is the use of multiple classifiers trained on different subsets of the data and a final classification being given with a voting scheme.

For noisy datasets bagging generally performs better than boosting and random forests [Die00, GBS06]. Other work reported on the use of boosting, bagging and random forests in remote sensing include [HCCG05, MB05, KN07].

The author believes that the use of different classification algorithms is a major strength of the work introduced in this thesis. Different classification algorithms use different properties of the training data and the combination of approaches increases the classification accuracy over just using a single classification algorithm.

In addition the classification framework used in this thesis is general enough to incorporate the use of boosting, bagging, random forests or other resampling techniques. However, the additional overhead in training may not result in further decreases in error rates. Investigation of this would certainly be worthwhile.

Overall multi-strategy approaches, such as that described in this thesis, continue to be widely reported in the literature and are preferred for small, noisy datasets.

C.2 Attribute Selection

Ensuring that the set of attributes used in a classification problem are relevant continues to be discussed in the machine learning literature. In the past attribute selection in remote sensing domains was carried out by domain experts based on their knowledge and / or detailed analysis of the data. It was not common to see more formal and rigorous methods of attribute selection being carried out in remote sensing and this does not appear to have changed very much.

The most commonly used attribute selection techniques are referred to as filter or wrapper techniques. Filter attribute selection uses a separate process that is independant of the classifier to determine the most relevant attributes. Approaches used include correlation of the attributes with the target outputs, or principle components analysis. Wrapper techniques were discussed in the thesis and were considered preferable as they allow the classifier itself to determine which are the most relevant attributes.

In 2003 NIPS ran a feature selection challenge [GGBD05]. One of the reasons for this was that the last decade has seen a number of application domains emerge that have very large numbers of attributes (up to hundreds of thousands) and yet can have small training datasets. One of the original motivations for the work carried out in this thesis was indeed the limited training data available in remote sensing, and this problem appears to have only compounded in the time since the original submission of this work. This indicates an increased need for classification frameworks such as that described in this thesis, and the continuing relevance of this kind of research.

Datasets with very large numbers of attributes, and limited training data were chosen for the NIPS challenge. Five datasets from different application domains

were used. All datasets were given as two-class classification problems. The data were split into three subsets – a training set, a validation set, and a test set. The identity of the data was concealed using a number of preprocessing techniques, as well as including additional attributes (called probes) similar to the real data but which contained no information. Participants could submit prediction results for ranking over a 12 week period.

The results were ranked based on four measures:

- The balanced error rate (BER), which is the average of the error rate of the positive class and the error rate of the negative class.
- Area under the ROC curve (AUC). The ROC curve is obtained by varying a threshold on the outputs of the classifier. The curve represents the fraction of true positive as a function of the fraction of false negative. For binary classifiers, $BER = 1 - AUC$.
- The fraction of features selected.
- The fraction of probes that were found in the attribute set selected.

The winners of the challenge used a combination of Bayesian neural networks [Nea96] and Dirichlet diffusion trees [Nea01] with attribute selection carried out as follows:

1. The number of attributes were reduced to a few hundred, either by selecting a subset of attributes using simple univariate significance tests, or by principal component analysis performed on all available labelled and unlabeled data.
2. They then applied a classification method based on Bayesian learning, using an automatic relevance determination prior that allows the model to determine which of the attributes are most relevant.

Common features of the solutions used by the participants were:

- Principle components analysis was used successfully by a number of the participants to reduce the number of attributes and did not require any domain specific knowledge to do so.

- Multi-strategy classification techniques were commonly used, with the winners and several of the top ranked solutions using combinations of methods. Some entries used voting to determine the final decision and these solutions gave improved classification results.
- Preprocessing of the data to generate additional attributes was used by a number of participants.

As wrapper methods are computationally expensive, filter methods have been used in preference for many years even though not considered as effective [GGBD05]. However, some of the top ranked entries in the feature selection challenge used one or more filters to reduce the number of attributes. As filters do not remove redundant attributes some solutions combined filters with other methods to reduce the number of attributes further and remove redundancies.

A conclusion from the challenge was that eliminating meaningless attributes was not critical – it is still possible to generate good classifications with the attributes that contain no information (i.e. the probes). The authors of the paper also state that a surprising result was that some of the best entries used all of the attributes, however, there were always entries that had only a small number of attributes that had similar error rates. It is still desirable to reduce the number of attributes to reduce training and classification times. Multi-strategy approaches that include attribute selection, such that used in this thesis, can be used to improve classification accuracy.

C.2.1 Neural Network Attribute Selection

A significant part of the original contribution in this thesis was the development of a neural network attribute selection technique. Many approaches to attribute selection for neural networks have been discussed, but very little that is directly comparable to contribution analysis.

The main approach that continues to be commonly used is to select attributes are filter approaches, such as principle components analysis or statistical significance tests, and then training the network on the resulting attribute subset. A sample of papers that use such approaches are [ANHN$^+$00, JA03, SS03, SR07].

Not much work could be found that discussed using the properties of the trained

neural network directly to determine the most relevant set of attributes (also referred to as embedded techniques) for use in classification. Very few entries in the NIPS feature selection challenge [GGBD05] used embedded techniques to determine the most relevant set of attributes.

Gascaa et al. [GSA06] proposed a technique similar to contribution analysis that uses the connection weights in multi-layer perceptron neural networks trained with back-propagation to determine the relevance of attributes.

The contribution of an input node i to output node o was defined as

$$C_{io} = \frac{\sum_{j=1}^{nhidden} w_{jo}\beta_j w_{ji}}{\sum_{l=1}^{ninputs} |\sum_{j=1}^{nhidden} w_{jo}\beta_j w_{jl}|}$$

where w_{ij} is the connection weight between input node i and hidden node j, w_{jo} is the connection weight between hidden node j and output node o, $ninputs$ and $nhidden$ are the number of input and hidden nodes respectively. The contribution of the hidden layer j is approximated by β_j as follows

$$\beta_j = (1/n)\sum_{t=1}^{n} O_{jt}(1 - O_{jt})$$

where n is the number of training cases and O_{jt} is the output of hidden layer j for case t of the training set.

The paper is not clear on what is meant by the output of a hidden layer and how this is calculated or determined. Neither is approximating the contribution of hidden layers to the outputs necessary. It is possible to calculate a specific contribution using the weights of the nodes in the hidden layers, the method used in this thesis being at least one example.

Once the contributions were calculated, the prominence of an input was then determined as follows.

1. Calculate the contribution of each input node to each output node
2. Sort the contributions for each output node in descending order
3. For each output node compute how many times (n_{il}) the ith attribute is in position l
4. The prominence P_i of attribute i is $\sum_{l=1}^{ninputs} \frac{1}{2^i} n_{il}$
5. Sort the attributes according to their prominence

The correlation between two attributes a and b was determined:

$$r_{ab} = \frac{c_{ab}}{\sqrt{c_{aa} c_{bb}}}$$

where

$$c_{ab} = (1/n) \sum_{i=1}^{n} (x_a^t - \mu_a)(x_b^t - \mu_b)$$

and where x_a^t is the value of attribute a for case t of the training set, and μ_a is the expected value.

Two attributes were correlated if $|r_{ab}| > 0.707$ and two methods for removing them were investigated:

Opt-I Discard the attribute with the lowest prominence value.

Opt-II Discard the attribute with the lowest prominence value provided that it is among the $ninputs/2$ top ranked attributes.

Nine of the benchmark databases from the UCI Machine Learning Database Repository (`http://www.ics.uci.edu/mlearn`) were used to test this method of attribute selection. The Opt-I approach gave higher error rates than both the Opt-II approach and using the complete dataset for classification. This was due to Opt-I removing relevant attributes from the dataset. The Opt-II approach gave

slightly lower error rates over 60% of the datasets used. Where Opt-II performed worse than using the entire dataset the error rate was only slightly higher.

The attribute selection method described by Gascaa et al. had the effect of removing redundant attributes, while contribution analysis, as discussed in this thesis, is better for removing irrelevant attributes. It is unclear how effectively contribution analysis performs in removing redundant attributes. It is certainly dependant on how the neural network uses the attributes - if the neural network had strong connection weights for redundant attributes they would be retained. Further investigation of this would need to be carried out to determine the ability of contribution analysis to identify redundant attributes.

The conclusion of Gascaa et al. was that attribute selection was most useful for reducing data dimensionality. Overall opinion in the attribute selection literature seems to now indicate that the removal of redundant or irrelevant attributes will contribute more to reducing computational load rather than reducing error rates. This thesis used attribute selection in combination with a number of other techniques to increase classification accuracy, which also agrees with the findings in [GGBD05].

C.2.2 Wrapper Attribute Selection in Remote Sensing Domains

At the time of writing the use of wrapper attribute selection was not apparent in remote sensing literature, although widely used in machine learning domains. Discussion of the application of wrappers to remotely sensed data continues in the machine learning literature, and is now also evident in the remote sensing literature [KM02, Yu03, VOVS05, BGK06, ZZW08].

C.3 Attribute Generation

Attribute generation has been discussed for some time across many application domains, including remote sensing. Much of the work done in explicitly generating attributes for use in classification, in the way that was carried out in this thesis, appears in the genetic algorithms literature and increasingly in the data mining literature.

In remote sensing attribute generation has generally taken the form of pre-processing the data, and then using the transformed data directly for classification. For example, generating an unsupervised classification and then mapping the labels to known classes, or to use attributes generated by principle components analysis as inputs to a classifier. This continues to be the case in remote sensing, and more rigorous and formal approaches for attribute generation, as was done in this thesis, are not apparent in the literature. The work presented in the data mining literature seems to indicate that this continues to be a productive area of focus for research.

More recent work investigating generation of attributes have been reported in [LX01, GM02, MR02, GGBD05, CAT05, SB05, MM05, IGW06]. Much of this work also includes the use of multi-strategy classification schemes and attribute selection.

C.4 Simulating Remotely Sensed Data

In the last few years remotely sensed data has certainly become more readily available on the internet. However, the problem of the availability of high quality classified data remains – the collection of ground truth data continues to be expensive and error prone [Yu03]. As a result the use of multi-strategy techniques, including some kind of attribute generation and selection, will continue to be relevant in this domain.

C.5 Citations Since 2000

Work that cited publications from this thesis since 2000 are given below. In particular, attribute selection for neural networks using contribution analysis continues to be relevant in the literature.

L.K. Milne, T.D. Gedeon, and A.K. Skidmore. Classifying dry sclerophyll forest from augmented satellite data : Comparing neural network, decision tree and maximum likelihood. In *Proc. 6th Australian Conference on Neural Networks*, Sydney, pages 160–163, February 1995.

Cited by:

Y. M. Sebzalli and X. Z. Wang. Knowledge discovery from process operational data using PCA and fuzzy clustering. *Engineering Applications of Artificial Intelligence*, 14(5):607–616, October 2001.

L.K. Milne. Feature selection using neural networks with contribution measures. In *AI'95 Poster Proceedings*, Canberra, November 1995.

Cited by:

D.F. Millie, G.R. Weckman and R.J. Pigg. Modeling phytoplankton abundance in Saginaw Bay, Lake Huron: Using artificial neural networks to discern functional influence of environmental variables and relevance to a great lakes observing system. *Journal of Phycology*, 42(2):336–349, April 2006.

D.F. Millie, G.R. Weckman and H.W. Paerl. Neural net modeling of estuarine indicators: Hindcasting phytoplankton biomass and net ecosystem production in the Neuse (North Carolina) and Trout (Florida) Rivers, USA. *Ecological Indicators*, 6(3):589–608, August 2006.

J.J Montano and A. Palmer. Numeric sensitivity analysis applied to feedforward neural networks. *Neural Computing & Applications*, 12(2):119–125, November 2003.

S. Piramuthu. On learning to predict Web traffic. *Decision Support Systems*, 35(2):213–229, May 2003.

S. Piramuthu. Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156(2):483–494, July 2004.

S. Piramuthu. On preprocessing data for financial credit risk evaluation. *Expert Systems with Applications*, 30(3):489–497, April 2006.

M.J. Watts and S.P.Worner. Using artificial neural networks to determine the relative contribution of abiotic factors influencing the establishment of insect pest species. *Ecological Informatics*, 3(1):64–74, January 2008.

C.6 Conclusion

The literature related to this thesis, published between 2000 and 2008 was reviewed. Publications discussing the key themes of

- multi-strategy classification
- attribute generation
- attribute selection

were reviewed, with a particular focus on work done using remotely sensed data.

Classification schemes that use more than one technique or classifier continue to be commonly reported as providing reduced error rates over just using a single classifier. Most approaches tend to use one or two techniques in combination, for example, attribute selection with a single classifier algorithm, or ensembles of a single classifier type to improve classification accuracy. No work was found that used the combination of attribute generation and selection, with a range of different classification algorithms.

Attribute selection also continues to be widely discussed. Many attribute selection techniques used are still computationally expensive and are typically based on some kind of brute force search or highly iterative approach. Neural network attribute selection using contribution analysis is still very relevant as it is an efficient means of removing attributes that are not used by the network, and so contain large amounts of noise or no information that the network can use. The relevance of this work is supported by the fact that the paper discussing contribution analysis continues to be cited in the literature.

There has been an increase in the discussion of explicit attribute generation techniques. This is now more an exercise in data mining, similar to the approach taken in this thesis, rather than just preprocessing the data. This is also often reported in combination with attribute selection.

In conclusion, the literature continues to support the relevance and original contribution of the thesis.

Bibliography

[ACN07] M. Awad, K. Chehdi, and A. Nasri. Multicomponent image segmentation using a genetic algorithm and artificial neural network. *IEEE Geosci. Rem. Sens. Letters*, 4(4):571–575, 2007.

[ANHN+00] W. Al-Nuaimya, Y. Huanga, M. Nakhkasha, M.T.C. Fanga, V.T. Nguyenb, and A. Eriksenc. Automatic detection of buried utilities and solid objects with GPR using neural networks and pattern recognition. *J. App. Geophysics*, 43(2-4):157–165, 2000.

[BGK06] A. Blansch, P. Ganarski, and J.J. Korczak. Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, 27(11):1299–1306, 2006.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[CAT05] S. Chebrolua, A. Abraham, and J.P. Thomas. Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, 24:295–307, 2005.

[CCP07] C. Cea, J. Cristobal, and X. Pons. An improved methodology to map snow cover by means of Landsat and MODIS imagery. In *IEEE Geoscience and Remote Sensing Symposium*, pages 4217–4220, Barcelona, July 2007.

[Die00] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization. *Machine Learning*, 40(2):139–157, 2000.

[DWBS+04] M. A. Diuk-Wasser, M. Bagayoko, N. Sogoba, G. Dolo, M. B. Tour, S. F. Traor, and C. E. Taylor. Mapping rice field anopheline breeding habitats in Mali, West Africa, using Landsat ETM+ sensor data. *Int. J. Rem. Sens.*, 25(2):359–376, 2004.

[GBS06] P.O. Gislason, J.A. Benediktsson, and J.R. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300, 2006.

[GGBD05] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the NIPS 2003 feature selection challenge. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 545–552. MIT Press, Cambridge, MA, 2005.

[GM02] G. Gomez and E.F. Morales. Automatic feature construction and a simple rule induction algorithm for skin detection. In *In Proc. of the ICML Workshop on Machine Learning in Computer Vision*, pages 31–38, 2002.

[GSA06] E. Gascaa, J.S. Snchezb, and R. Alonsoa. Eliminating redundancy and irrelevance using a new MLP-based feature selection method. *Pattern Recognition*, 39(2):313–315, 2006.

[HCCG05] J. Ham, Y. Chen, M.M. Crawford, and J. Gosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Rem. Sens.*, 43:492–501, 2005.

[Ho98] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[IGW06] R. Islamaj, L. Getoor, and W.J. Wilbur. A feature generation algorithm for sequences with application to splice-site prediction. *Lecture Notes in Computer Science, Knowledge Discovery in Databases: PKDD 2006*, Vol 4213/2006:553–560, 2006.

[JA03] J. Jerez-Aragon. A combined neural network and decision trees model for prognosis of breast cancer relapse. *Artificial Intelligence in Medicine*, 27(1):45–63, 2003.

[KKOL00] S. Kang, S. Kim, S. Oh, and D. Lee. Predicting spatial and temporal patterns of soil temperature based on topography, surface cover and air temperature. *Forest Ecology and Management*, 136(1):173–184, 2000.

[KM02] T. Kavzoglu and P. M. Mather. The role of feature selection in artificial neural network applications. *Int. J. Rem. Sens.*, 23(15):2919–2937, 2002.

[KN07] S. Kawaguchi and R. Nishii. Hyperspectral image classification by bootstrap adaboost with random decision stumps. *IEEE Trans. Geosci. Rem. Sens.*, 45:3845–3851, 2007.

[LCJM04] B. Liu, Q. Cui, T. Jiang, and S. Ma. A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinformatics*, 5(136):139–157, 2004.

[LE07] J. Lee and O.K. Ersoy. Consensual and hierarchical classification of remotely sensed multispectral images. *IEEE Trans. Geosci. Rem. Sens.*, 45(9):2953–2963, 2007.

[LVK+05] Y. Li, A. Vodacek, R.L. Kremens, A. Ononye, and C. Tang. A hybrid contextual approach to wildland fire detection using multispectral imagery. *IEEE Trans. Geosci. Rem. Sens.*, 43(9):2115–2126, 2005.

[LX01] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. *IEEE Symposium on Security and Privacy*, 00:0130, 2001.

[MB05] C. Mingmin and L. Bruzzone. A semilabeled-sample-driven bagging technique for ill-posed classification problems. *IEEE Trans. Geosci. Rem. Sens.*, 2(1):69–73, 2005.

[MM05] I. Mierswa and K. Morik. Automatic feature extraction for classifying audio data. *Machine Learning*, 58(2-3):127–149, 2005.

[MR02] S. Markovitch and D. Rosenstein. Feature generation using general constructor functions. *Machine Learning*, 49(1):59–98, 2002.

[Nea96] R.M. Neal. Bayesian learning for neural networks. *Number 118 in Lecture Notes in Statistics*, pages –, 1996.

[Nea01] R.M. Neal. Defining priors for distributions using dirichlet diffusion trees. Technical Report Technical Report 0104, Dept. of Statistics, University of Toronto, March 2001.

[SB05] M.G. Smith and L. Bull. Genetic programming with a genetic algorithm for feature construction and selection. *Genetic Programming and Evolvable Machines*, 6(3):93–100, 2005.

[SMS00] J.J Simpson, T.J. McIntire, and M. Sienko. An improved hybrid clustering algorithm for natural scenes. *IEEE Trans. Geosci. Rem. Sens.*, 38(2(2)):1016–1032, 2000.

[SR07] R.K. Sivagaminathana and S. Ramakrishnan. A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert Systems with Applications*, 33(1):49–60, 2007.

[SS03] R.W. Swiniarski and A. Skowron. Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, 24(6):833–849, 2003.

[VOVS05] C. Vaiphasa, S. Ongsomwang, T. Vaiphasa, and A.K. Skidmore. Tropical mangrove species discrimination using hyperspectral data: A laboratory study. *Estuarine Coastal and Shelf Science*, 65(1-2):371–379, 2005.

[Yu03] S. Yu. *Feature Selection and Classifier Ensembles: A Study on Hyperspectral Remote Sensing Data*. PhD thesis, University of Antwerp, School of Computer Science and Engineering, Antwerp, Belgium, 2003.

[ZZW08] P. Zhong, P. Zhang, and R. Wang. Dynamic learning of smlr for feature selection and classification of hyperspectral data. *IEEE Geoscience and Remote Sensing Letters*, 5(2):280–284, 2008.