Data Mining as a tool for investigating Environmental Systems

JM Spate

October 1, 2006
# Contents

1 Data and Problems .................................................. 1
   1.1 Introduction ....................................................... 1
   1.2 Environmental Data .............................................. 4
      1.2.1 Rainfall .................................................... 4
      1.2.2 Streamflow ................................................. 5
      1.2.3 Temperature ................................................. 7
      1.2.4 Evaporation, Potential Evaporation, and Evapotranspiration ... 8
      1.2.5 Relative Humidity .......................................... 9
      1.2.6 Radiation and Sunshine Hours ................................ 9
      1.2.7 Wind Speed .................................................. 9
      1.2.8 Chemical Concentrations, Solute, and Suspended Sediments ... 10
      1.2.9 Boolean and Categorical Variables .......................... 10
      1.2.10 Spatial Data ................................................ 11
      1.2.11 Metadata .................................................... 12
      1.2.12 Other Data Issues to Consider ............................ 13
   1.3 Hydrological Modelling ........................................... 14
      1.3.1 For What Purpose? ......................................... 16
   1.4 Further Chapters .................................................. 17

2 Review of Data Mining Techniques .................................... 23
   2.1 Introduction ....................................................... 23
   2.2 Key Data Mining Concepts and Definitions ...................... 25
   2.3 Clustering .......................................................... 26
   2.4 Classification .................................................... 28
      2.4.1 An Example .................................................. 29
      2.4.2 C4.5 Decision Trees ....................................... 31
2.4.3 A Consideration: Data Types, Missing Values, Noise ........................................... 35
2.4.4 Pruning and Tree Sizes ............................................................................................... 36
2.4.5 Attribute Selection, Bagging, and Boosting ............................................................... 37
2.4.6 Class Distribution and Resampling ........................................................................... 40
2.4.7 Benchmarking ............................................................................................................ 42
2.4.8 Other Algorithms ...................................................................................................... 42
2.4.9 Evaluating Classifier Performance, and Cost Matrices ............................................ 47

2.5 Rule Extraction ............................................................................................................. 50
  2.5.1 Rules ......................................................................................................................... 50
  2.5.2 Apriori- an Example Algorithm ............................................................................... 52
  2.5.3 A Second Rule Extraction Algorithm- ITRULE ..................................................... 53
  2.5.4 Rule Selection ......................................................................................................... 54

2.6 Similarity Measures ..................................................................................................... 54
  2.6.1 Discretisation .......................................................................................................... 55
  2.6.2 Text Distance ......................................................................................................... 55
  2.6.3 Data Reduction by Extraction of Dominant Modes ............................................... 57

2.7 Previous Applications .................................................................................................. 59

2.8 Conclusions .................................................................................................................. 61

3 Vulnerability to Gully Erosion ...................................................................................... 63
  3.1 Statement of the Problem, Motivation, and Data ........................................................... 63
    3.1.1 Data ....................................................................................................................... 64
  3.2 Approach- Gully Density ............................................................................................. 67
  3.3 Visual Investigation ...................................................................................................... 69
  3.4 Application of Data Mining Algorithms ........................................................................ 71
  3.5 Discussion of Results .................................................................................................. 74
  3.6 Comparison with Classical Techniques ....................................................................... 77
  3.7 Conclusions ................................................................................................................ 78

4 Rainfall Intensity ............................................................................................................ 81
  4.1 Statement of the Problem and Motivation .................................................................... 82
  4.2 Training Data .............................................................................................................. 83
    4.2.1 Data Quality ......................................................................................................... 85
  4.3 Definition of an Intense Event ...................................................................................... 87
    4.3.1 Clustering Rainfall Events ................................................................................... 87
CONTENTS

4.3.2 Other Classifiers .................................................. 90
4.4 Classification Algorithms ........................................... 92
4.5 Results of Classification using C4.5 and ADtree ............... 96
4.5.1 Results of Cnt2 Classification ................................ 96
4.5.2 Results of Int, Cnt, and Cnt3 Classifications ................ 98
4.5.3 Parameter Selection for C4.5 and ADtree Algorithms ........ 99
4.5.4 Attribute Selection ................................................. 102
4.5.5 Additional Experiments ......................................... 107
4.5.6 Regionalisation ..................................................... 110
4.6 Conclusions ........................................................... 112
4.7 Further Work ........................................................ 113

5 Modelling Water Quality .................................................. 115
5.1 Data ........................................................................... 117
5.2 A Brief Discussion of Noise in Flow and Water Quality Data .. 120
5.3 A Brief Note on Simple Models .................................... 121
5.4 Data Mining Approach and Visualisation ......................... 122
5.4.1 Automated Shape Extraction .................................... 129
5.5 Model Structure ......................................................... 131
5.5.1 Preliminary Model Trees ......................................... 132
5.5.2 Model Trees, Error and Uncertainty .......................... 134
5.5.3 Season Definition Experiment ................................... 137
5.6 Regression with Additional Features ............................... 138
5.6.1 Results from Regression Experiments ....................... 142
5.7 Conclusions ........................................................... 143

6 Modelling Electrical Conductivity ...................................... 145
6.1 Introduction ............................................................. 145
6.2 Description of Available Data ...................................... 147
6.3 Data Structure ........................................................ 149
6.4 Rule Extraction ......................................................... 150
6.5 Frequent Subsequences .............................................. 152
6.6 Quantifying Characteristic Shapes ................................. 154
6.7 Two-dimensional Relationships .................................... 155
6.8 Conclusions ........................................................... 158
7 Clustering Time Series

7.1 Introduction .................................................. 161
7.2 Distance Measures and Metrics ................................. 163
7.3 Basic Investigation ........................................... 165
  7.3.1 Stability Considerations and Geographical Clustering .... 165
  7.3.2 Missing Distance Estimation ............................... 168
7.4 Distances between Series .................................... 169
  7.4.1 Mutual Information as Similarity Measure ............... 170
7.5 Distance Measures between Series ............................ 175
7.6 Hierarchical Clustering ....................................... 176
7.7 Clustering with Frequent Patterns ............................ 180
  7.7.1 Verification of Method ................................... 181
  7.7.2 Results from South-East Queensland Flow Series ....... 185
7.8 Conclusion .................................................... 186

8 Plynlimon .......................................................... 189

8.1 Introduction .................................................... 189
8.2 Data Quality and Preprocessing ................................. 192
8.3 Characteristic Motifs within the Hafren Dataset ............. 193
8.4 Clustering the Hafren Time Series ............................. 197
  8.4.1 Clustering the Hafren Time Series with Motif Distance ... 197
  8.4.2 Clustering the Hafren Time Series with Euclidean Distance 200
8.5 Comparison of Variables Across Locations .................... 202
8.6 Conclusions .................................................... 205
8.7 Further Work .................................................. 206

9 Conclusions ....................................................... 207

9.1 Introduction .................................................... 207
9.2 Review of Data Mining Techniques ............................. 207
9.3 Gully Erosion Location Classification ........................ 208
9.4 Rainfall Intensity Extraction .................................. 208
9.5 Jugiong, Muttama and Tarcutta Water Quality ............... 209
9.6 Electrical Conductivity Modelling ............................. 209
9.7 Flow Time Series Clustering ................................... 210
9.8 Plynlimon Series Investigation ................................. 212
9.9 General Conclusions ........................................ 212
Acknowledgements

First, I would like to thank my supervisors Tony Jakeman and Barry Croke of the Integrated Catchment Assessment and Management Centre (iCAM), Steve Roberts of ANU Mathematical Sciences Institute, and John Gallant of CSIRO Department of Land and Water, without whom this work would not be possible. In particular, Barry’s boundless enthusiasm for new ideas and stunning intelligence were invaluable, as was Tony’s care and attention when editing papers and chapters. I would also like to thank him for all the vinyl, and for ‘No Sleep Til Belfast’. Rebecca Letcher of iCAM has been a pillar of strength and wisdom since before my honours year, especially in the last few months.

Over the last three years my collaborators have been too numerous to mention. Celina Smith of iCAM, who generously provided the data used in Chapters 3 and 5, deserves a special mention, as does Karina Gibert of the Technical University of Catalonia. I look forward to working more with her, and my colleagues Quim Comas and Miquel Sánchez-Marrè in the future.

My family, especially my mother Jan, have provided patience, interest, support, and meals, often when most needed and least expected. I doubt I will ever be able to repay such kindness, nor express my gratitude as deeply as I feel it.

I would like to thank Fil Giles and Armando Corvini for beautiful days on rock, and Canberra Indoor Rock Climbing Centre for being my home away from home.

To my comrades, who share the belief that we have a world to win, I extend my solidarity and admiration. Ben Halliday and Ben Keaney are among those who showed me how the struggle should be taken forward, and Tanya McConvell convinces me every time I see her that it can be won.

The Dropkick Murphys, Strike Anywhere, Against Me!, Tiger Army, H-Block 101, The Disables, Final Warning, The Porkers, Stiff Little Fingers, Los Capitanes, Unpaid Debt, and every other live band I have seen in these three years make the world an better place to live in every time they play.
Throughout the construction of this thesis, my friends have been there every time I turned around. To Jim, Patricia, Ahmed, Chris, Kes, Nikki, Ry, Cam, Mary, Jade and all the others, thankyou. I hope you had as much fun as I did. There is one friend conspicuous by her absence in the above list - Claire Carpenter. Without you I am lost, now as always.
Declaration of Originality

Except where appropriately attributed, this thesis is my own work. It has not previously submitted to any institution of higher education for any degree or diploma.

Some parts of Chapter 2 are shortly to be published as part of the proceedings of the International Environmental Modelling and Simulation Society 2006 Conference, as an invited position paper co-authored by K Gibert, M Sánchez-Marrè, E Frank, J Comas, I Athanasiadis, and R Letcher. A preliminary study of the problem detailed in Chapter 4 was published as part of my Honours thesis, submitted to the this university in 2002, and developed in a paper published as part of the proceedings of the 2003 Modelling and Simulation Society of Australia and New Zealand conference. This paper was co-authored by BFW Croke and AJ Jakeman. Parts of Chapter 6 have been published in the proceedings of the 6th International Symposium on Intelligent Data Analysis (2005). The literature review of Chapter 2 has been submitted, in modified form, to the Environmental Modelling and Software journal with a view to publication.

Signed: [Signature]

Date: [Date]
Chapter 1

Environmental Data and Environmental Problems

1.1 Introduction

In this thesis we will consider the uses of data in the context of environmental modelling, and strategies for extracting the maximum amount of useful information from it. Environmental data have some particular characteristics that may hamper the knowledge extraction task unless they are dealt with properly. Most of these issues stem from the complexity of the systems under consideration, characterisation of which are difficult. Available data may also be limited in both quantity and quality. Thus, efficient use of that data which does exist is of greatest interest and importance.

Useful information can be extracted directly from data at the preprocessing stage, where visualisation and other tools can not only be used to investigate the system under consideration, but also identify potential problems and other issues for further modelling. The more automated analysis techniques of data mining can then be applied in an intelligent and focussed manner, to obtain the best possible quality of result without prohibitive manual work. The investigation and presentation of results in a manner interpretable to humans is also an issue that must be addressed if those results are to be applied in the real world.

Data mining techniques have enjoyed a rapid rate of development over the last two decades, primarily due to increased rates of data collection in many sectors and massive rises in computation power. For example, the use of scannable bar code technology in supermarkets has led to the creation of databases of Terabytes in size, and the technology to
store and process this data digitally now exists. Rule extraction algorithms (for example [3]) have been developed to discover trends within these datasets that may be of use to retailers and manufacturers. Modelling in the environmental sciences has largely evolved within a paradigm where the emphasis has lain on the assumption and codification of certain prior knowledge about the system, rather than direct investigation of data. There is no reason why the environmental sciences should not benefit from algorithms and paradigms developed for analysis of the datasets of other disciplines.

Data mining algorithms are powerful tools for investigating large datasets, and throughout this project we aim to illustrate the ways in which they can be applied, taking into account the particular issues that surround environmental data and environmental modelling. We consider data preprocessing, treatment of uncertainty, model and parameter selection, performance reporting and presentation of results on an equal footing with the modelling algorithms themselves. These steps are essential steps in the best practice modelling process for any problem. In environmental modelling, where data is often of relatively poor quality, they are all the more important.

Rigorous illustration of data mining techniques that extract the maximum amount of information for a specified purpose should serve as an example of the kind of analysis that can be performed with these techniques, as well as being useful in its own right. In later chapters, we perform a number of experiments towards both of these aims.

It is hoped that better use of data will assist in leading to improved models and associated predictions for problems that arise in environmental science and management. Modelling needs are many and diverse: we will consider a range of problems including rainfall data downscaling, pollutant and solute transport, identification of spatial locations vulnerable to gully erosion, and flow series characterisation.

Environmental modelling is set apart from other disciplines by a number of factors (aside from the potential importance of results and findings). One of the most significant of these is the nature of available data. In most cases, controlled experiments are impossible and data can only be obtained by measuring the system in its present state. It is not in general possible to change any one factor and observe the result.

Almost all environmental systems are spatial in nature, and although it may be possible to model the desired variables in one or two dimensions, other variables will certainly vary spatially. Consider for example the rainfall-flow process for a given catchment. While output flow takes the form of a time series, incident rainfall will not be constant over the whole catchment area (unless the catchment is extremely small). Soil types, vegetation,
and a host of other important quantities are also unlikely to remain constant. Capturing such spatially varying features at an appropriate resolution is difficult and expensive, and estimation is often used in the place of direct measurement. It is also common practice to aggregate a number of spatial samples into one, representing the catchment as a whole. Several rainfall measurements from different locations are often combined into one catchment-wide value, for instance.

Evolution and variation in time are often the focus of environmental models. In many cases, such change over time is of paramount importance to industrial production or environmental sustainability. In the time dimension, there are two restrictions on available data: firstly, attainable frequency of sampling, and secondly, existence of past records. The former issue is similar to that of spatial resolution, governed by costs, scale of variation, and ease of sampling. The latter is more difficult, and may again force modellers to resort to estimation, or use of lower quality, extrapolated, or qualitative data to fill the gaps in the knowledge base.

Extrapolation from multiple sources, low resolution sampling, and measurement difficulties all introduce uncertainty into the dataset (see Section 1.2.11). Highly uncertain data is a pronounced feature of environmental problems, and is not always considered with due attention. It is not uncommon that the issue is ignored wholesale, even though the impact of input uncertainties on results may be extensive.

Outliers are also common. These are points with anomalously high or low values. They may be a result of a measurement or transcription error, or a true feature of the system. For example, in the study described in [143], a small number of rainfall events an order of magnitude larger than typical were found. Investigation revealed that some of these were caused by misplacement of the decimal point when the data were manually digitised, but others were the physical result of a tropical cyclone that passed over the area. Outliers can have a strong effect on final results, and so must be carefully identified and treated.

Data is the foundation of most environmental models. Section 1.2 discusses specific variables that may be used when building environmental models, and concerns that may arise from them. Section 1.3 provides a short summary of the field of hydrological modelling, which covers several examples discussed in later chapters. It is useful to consider some of the aims of this kind of work, common techniques in use, and the primary difficulties encountered, as a guide to what can be expected in the broader field. We then move on to discuss the projects undertaken as part of this thesis in Chapter 1.4.
1.2 Environmental Data

As in later chapters, we should first consider the nature of the data available to us. Here we shall consider some of the variables that are used in environmental science, and some issues that typically arise from them. This section is intended as a reference for subsequent projects, and also to provide an introduction to some of the characteristics of specifically environmental data.

1.2.1 Rainfall

Australian rainfall data is usually taken in the form of a series of daily totals from 9am to 9am, often at post offices, airports, or by individual farmers. Measurements at sub-hourly intervals are made at a small number of sites, and a handful of records are created with 'tipping buckets' where a time stamp is noted each time a set bucket volume is exceeded. The bucket is then emptied and filled again, providing a variable-resolution rainfall series. However, daily totals make up the overwhelming majority of all rainfall records, and most of these are recorded by hand. Also note that some confusion may occur when the rainfall from 9AM yesterday to 9AM today is recorded and labelled with a date. Most other variables consider a day to start and end at midnight.

Daily rainfall data alone usually gives little indication of event intensity. An average record of a few tens of millimetres may fall evenly over the course of the whole day, or in one burst of a few minutes duration. The amount of runoff from an intense event tends to be much larger than that from a gentler event of the same total magnitude, and also mobilises and carries a very different amount of pollutants (eg fertilizers), other solutes, and suspended material with it. Rainfall values for a specific catchment or location are also often obtained by spatial interpolation, for example by thin plate smoothing splines (see for example [69]). As noted earlier, this process introduces additional uncertainty. The assumption that one value can adequately represent the rainfall incident on a large area may also be invalid.

Unless otherwise stated we, like most researchers in Australia, shall express rainfall values in millimetres, but it is not wise to assume that all rainfall records- even Australian records- are in millimetres. Inches and points (tenths of an inch) are not uncommon, and yearly totals are often expressed as centimetres or metres in wetter regions. The standard international units (Systéme International d'Unités or SI units) do not have the prevalence in environmental science that it does in, for example, physics.
Most rainfall-runoff modelling applications use daily data, chiefly because information at smaller time scales is simply not available, but intensity information could be integrated relatively easily into most models. This information has the potential to improve the predictive performance of the models for a range of purposes. For example [29] argue the need for discharge predictions at higher temporal resolution in order to better estimate sediment loads in Australian rivers.

Errors in daily point rainfall data are typically estimated at 5% to 10% on each measurement, and may be considerably higher for areal (spatially aggregated) estimates ([27]). From a basic rain gauge, water may escape or enter: by evaporation or condensation, splashing or animal intervention. Assuming an error of this magnitude, the danger of over-fitting models must be considered serious. Where higher resolution data is available, and it is collected at about 1000 stations Australia wide, the errors are lower. Gross errors at single points must also be considered (see discussion on outliers in the introduction to this chapter). Human error is always a possibility, especially with old, analogue gauges, and can manifest in the form of missing decimal points, mistimed readings, and lost or illegible measurements.

In the 1971 study [70], 20000 rainfall records from a variety of sources were examined. They found that 2.4% of the data points were suspect under visual examination, and 4.1% were flagged as likely to be erroneous by a simple automated analysis. While the quality of high-end rainfall measurement stations have improved considerably since 1971, the standard manual rainfall gauges and reading techniques have changed little. Most of the suspect values found in [70] could be replaced with other values, but we must be aware of the need for scrutiny of rainfall data in the preprocessing phase.

Particularly notable for rainfall records is the possibility of unmarked missing values. Like many Boolean variables (see Section 1.2.9), measurements may only be noted on days the gauge is checked, and assumed to be zero elsewhere. However, if the gauge is not checked for several days, rainfall may aggregate and an erroneous value be noted on the day in question, and zeros inserted for the previous days, where some rainfall did in fact occur.

1.2.2 Streamflow

Runoff (streamflow, the volume of water flowing in a river, creek, or other waterway) is also often measured daily, but can be taken at much smaller time intervals. Records with a resolution of minutes are not uncommon. Clearly, the problem of measuring streamflow per
unit time is non-trivial. Even in the simplest case, where water flows through a rectangular weir where the cross-sectional area is known, the rate of flow through the weir is not uniform (see for example [84]), although the flow profile and hence total volumes passing through can be estimated. Where the cross-sectional area is less simple, the calculation becomes correspondingly more difficult. Figure 1.1 shows an example flow profile through the centre and along the edge of a rough riverbed. Clearly, estimating the total flow through this channel would be complicated, as the velocity profile is not simple. Temporal variation should also be considered, as flow measurements may not be a daily aggregate but taken at a single point in time.

Perhaps the most elaborate streamflow measuring equipment, an Ultrasonic Doppler Instrument, has an error of at least a few percent in conditions where the relationship between a point velocity measurement and the complete flow profile is well known. See [98] for example technical specifications of such an instrument. Change in conditions such as temperature are also reported to cause failures or false readings in similar systems, as discussed in [1]. In 1998, there were only 100 of these systems or similar ones operating in the USA.

While it has been stated ([1]) that uncertainties as low as 5% can be achieved for streamflow measurement, most records are derived from instruments with much lower performance than this. Uncertainty in streamflow measurements should be considered as at least 10% ([27]), and due to the complexity of taking measurements, the errors should probably not be assumed to be normally distributed. However, we must remember that while we are modelling physical processes, in effect we are seeking to determine the true value by modelling the reading on the measurement device, as this represents our knowledge of the system. Thus, what we are actually modelling is the data that would be read in a particular situation.

The time of day at which flow is measured is relevant. The typical time of measurement for daily streamflow is 12 midnight (at least where measuring apparatus is automated). In Section 1.2.1 we noted that rainfall 's typically recorded from 9AM to 9AM. Thus, the definition of a streamflow 'day' and a rainfall 'day' differ, which may cause apparent anomalies when data is later examined. For example, the assumption of most daily rainfall-runoff models that the runoff resulting from a rainfall event will appear on or after the day the event is recorded, ie the water falls from the sky before it turns up in the river, may be violated. For more on the effects of this data phenomenon on modelling, and a suitable correction method, see [142].
Streamflow values may be expressed in cubic metres per second, commonly called cumecs, or the equivalent imperial unit cusecs, cubic feet per second. Water volumes are usually given in megalitres, or cubic metres (that latter unit are equivalent to kilolitres). For hydrological modelling, the millimetre-depth is often an important quantity. This is the volume in megalitres divided by the catchment area in square kilometres, and is the depth in millimetres to which the catchment area would be submerged by this volume (ie 1 ML = 1 mm over 1 square kilometre). Sources in the United States and India in particular use acre-feet, one acre-foot being the area of an acre by the depth of one foot. Any of these volume units may be (and often are) converted to a flow unit by dividing by one timestep, which is usually an hour, day, month or year.

1.2.3 Temperature

Four daily temperature measures are widely available. They are:

**Maximum** Highest temperature achieved through the day, measured above ground

**Minimum** Lowest temperature reached during the day, measured above ground

**Mean** The average of the above two readings

**Grass** Lowest temperature of the day, measured at ground level

Those temperature recordings made above ground are typically collected from instruments in a louvered box at 1.5 metres or so above ground. The boxes are designed to
minimize interference by wind, rain, or sun. For sites with sub-daily or continuous temperature sampling, mean temperature may also refer to a more accurately estimated daily mean.

The grass temperature is usually lower than the daily minimum, because of the influence of the cold ground mass ([82]). It is, however, much harder to keep a gauge at ground level clear of other interferences from dust, animal interaction, and plant matter. The standard site for the measurement of grass temperatures is a well-mown and watered lawn away from built-up structures or taller vegetation ([83]), and it is used in evapotranspiration calculations (see Section 1.2.4). The influence of snow on temperature readings is clearly significant in the case of grass temperature, but the altered ground profile also has an effect on the other three temperature types mentioned.

Daily temperature measures suffice to describe the temperature behaviour of most systems. Each of the temperature measures varies much less on a daily time scale and much more smoothly on broader time scales than rainfall or streamflow, and the measuring apparatus is more than adequate to capture information at an appropriate precision, as the measurement task is much easier. Thus we may assume that errors in daily temperature readings are much lower than for the variables described above, of the order of 1%, or negligible ([27]). All temperatures we will use are given in degrees Celsius.

### 1.2.4 Evaporation, Potential Evaporation, and Evapotranspiration

Potential evaporation, the amount of water lost from a well-watered surface, may be estimated using pan evaporation (water lost from a shallow pan) or calculated with an approximate formula such as the Penman-Monteith equation (see for example [53]) for evaporation from crops, using daily temperature, solar radiation, and other variables. Obtaining exact actual evaporation values is difficult by either method, and evaporation does not always vary smoothly in space due to the effects of soil water content and plant cover. However, it usually assumed that a rough estimate suffices for most modelling purposes. In many hydrological models, such as IHACRES ([43]), the effect of evapotranspiration is limited, and the errors incurred do not strongly affect the modelling process. Evapotranspiration is the name given to the total water lost from evaporation and from transpiration, the water removed from the surface by vegetation. This is of course impossible or near impossible to measure at the catchment scale, so estimates have to be obtained using experimental results and/or empirical formulae (see [53] for details). That the units here are usually
millimetres over a given area.

1.2.5 Relative Humidity

Relative humidity is described in [83] as the ratio between the amount of water present in the air and the amount of water that would be present at the point of saturation, and is expressed as a percentage. Other humidity measures can be substituted for relative humidity, such as vapour pressure and dew point, but these are less frequently recorded than the relative humidity. A number of different instruments, with very different philosophies, are used to obtain relative humidity estimates. They range from wet-bulb and dry-bulb thermometer sets (psychrometers), where the cooling due to evaporation is measured and the relative humidity inferred, to devices that use the expansion of some appropriate material with moisture, or changes in electrical resistance, to obtain a more direct estimate. The latter is more widely used today.

1.2.6 Radiation and Sunshine Hours

Solar radiation is infrequently recorded, but is useful for many purposes where it is available. The SI units here are mega-Joules per square metre. Sunshine hours are slightly more common in meteorological datasets, and are simply the number of hours the sun is visible within a given day. Measurements are sometimes taken with a Campbell-Stokes recorder (or 'crystal ball'), a glass ball which collects sunlight from all angles. A piece of paper underneath the ball is burnt by the rays when they are sufficiently strong.

1.2.7 Wind Speed

Daily wind speeds may be average values or gust maxima. It is prudent to be sure which one of these two possibilities your data represents (see Section 1.2.11). Note that wind speed is an important factor in the determination of evaporation values, and is difficult to measure well as it varies on short temporal and spatial scales, even with expensive apparatus which is rarely available. Several units (knots, metres per second, kilometres per hour) are commonly used in the recording of wind speeds, and one must be careful to treat units appropriately. Note that one knot is approximately equivalent to 1.85 kilometres per hour, or 0.53 metres per second. Winds may also be recorded as wind runs, a distance equivalent to the integral of wind speed over the sampling period, which is easily determined
by the number of turns of the anemometer (wind speed meter). Where possible, we will use SI units, which in this case are metres per second.

1.2.8 Chemical Concentrations, Solutes, and Suspended Sediments

Small elemental and chemical concentrations such as nitrate and iron can be expressed in parts per million or parts per billion. Salt loads are often expressed as an electrical conductivity, for example in units of Siemens per centimetre. These are typically collected in the field and measured in a laboratory, which may provide an estimate of uncertainty along with the final values. Suspended sediment levels may be given as a concentration or as absolute loads, and are extremely difficult to measure accurately (see Section 5 for more on this topic). They may also be measured indirectly by examining turbidity with a nephelometer, in which case they may be reported in Nephelometer Turbidity Units (NTUs). Acidity is also a common measurement, and is usually expressed on the pH scale.

1.2.9 Boolean and Categorical Variables

A number of Boolean presence/absence variables are common, particularly in climate data. The MetAccess database ([67]), for example, lists Thunder, Hail, Snow, Frost, Haze, Fog, Dust, Gale, and Strong Wind.

Spatial flora and fauna species data are also often expressed as Boolean indicators, either listed with a spatial tag or on a GIS grid. Particularly in the case of fauna presence data, it is important to note that a zero or unlisted value may not imply absence: the fact that no sighting was made does not necessarily imply that the species is not present. In the case of apparently systematic records such as the time series of daily thunder indicators, missing points are often marked as absence, and so are indistinguishable in the record from null entries. Thus there is the possibility that an entry that should be positive is marked as null (see page 48) in the data.

Gale and Strong Wind are defined in the MetAccess database as mean wind speeds of above 33 knots and 22 to 33 knots respectively ([137]). See the Wind speed section (Section 1.2.7) for the discussion of the knot as a unit.

The wind-related indicators above could also be considered as a single categorical variable with values low to mild, strong, and gale. Variables that are difficult to measure or quantify such as air and water quality, weed infestation, and species risk are often given
1.2. ENVIRONMENTAL DATA

in this ordered categorical format. Records may be generated by discretisation (see Section 2.6) of real-valued numerical data according to some natural boundaries, for example the human age in years may be discretised into child and adult by splitting at 18 years. Usefulness of such a re-characterisation would be determined by the application area. It may also be appropriate to discretise or lower the precision of noisy data to remove the effect of small variations of magnitude less than the estimated or theoretical uncertainty.

It is also quite common to use subjective judgement to create categorical data, either by direct observation of a process that is difficult to quantify or by combining several observations under an imprecise objective function. For example, most high quality modelling projects will produce several measures of model performance. Each of these should quantify a different aspect of fit to observed data. $R^2$ (efficiency) is a general pointwise measure of fit to a time series variable, and Bias quantifies the difference in areas under the curve of modelled and observed time series data, and many other statistics can be extracted as appropriate to the task. These can then be combined either by a concrete objective function or subjectively into a single performance measure.

A host of other environmental factors are best considered as categorical. Land usage, crop type, geological formation, and soil type are some of these. The distinction between categorical and ordinal data must also be made. Some data such as land use category have no natural order, ie a value of pasture is not greater than or less than a value of forestry, but in the wind speed example above, the values low to mild, strong, and gale are clearly ordered. Ordering may be an important factor when considering errors (as we shall see in Section 3.4). Confusion between wind classes strong and gale is less serious than confusion between low to mild and gale, for example.

1.2.10 Spatial Data

The science of Geographic Information Systems (GIS) has made analysis of large spatial datasets possible with comparative computational ease. See for example [88] for a general reference. Some of the environmental spatial variables we could consider include vegetation type, LANDSAT satellite images, slope, aspect (lie with respect to the sun), lithology (rock type, usually very coarse in scale), or mineral concentrations. These kinds of geographic variables may be expressed as covering polygons, a raster, isolated points, or vectors.

Altitude can be captured in Digital Elevation Models (DEMs), grids with latitude and longitude information implicit in the grid reference. DEMs are derived from satellite observations, aerial photography, airborne laser ranging, and other airborne sources. The
continent of Australia is covered by a DEM with a 9-second spacing ([68]). 9 seconds is approximately 250 metres, although of course the true distance varies with latitude. Significant sub-areas are covered by DEMs with a resolution of down to 25 metres in either direction, and some data is available with 10m resolution or even better. Current state of the art data collection methods can sample at sub-metre resolution, although this data is of course very expensive to collect.

In additional to providing navigational information, DEMs are used to generate relief (slope) data, flow accumulation from the sub-slopes in a given catchment, and other spatial data. Where this is done, proper collection of metadata is essential so that precision, uncertainty, and possible biases of the new variables can be assessed.

1.2.11 Metadata

By metadata, we mean the information describing the nature of a given dataset. For example, with a collection of temperature readings, in order for the information to be meaningful we need to know where and when the data was gathered, the type (max, min, etc) of reading, and the units. Some information on expected levels of precision is also desirable, as is information about recording apparatus and how missing points are flagged. With spatial data the need for metadata becomes even more profound. Required metadata, along with that suggested above, should include spatial sampling information and describe any interpolation process used to generate the GIS data in hand.

GIS datasets, especially those with a large spatial extent, must also have a projection listed. Are the coordinates in latitude or longitude, metres from a given point, or another unit? If map or Global Positioning System (GPS) coordinates are used, which map set is being referenced? Considerable differences exist between, for example, co-ordinates expressed in the AGD66 and GDA98 map datum for the same location, although both systems are widely used in Australia. Lack of projection metadata can cause serious (although simply preventable) problems in an emergency situation such as a rescue or bushfire operation as well as in a research context.

Unfortunately, metadata is often much neglected. This is a particular problem for wind speeds, rainfall totals, and distance units (although some other units are certainly similarly affected), as measurements in miles and kilometres, metres and yards and feet, millimetres and points, knots and kilometres per hour and metres per second will be of similar orders of magnitude, and thus it is not immediately clear which unit is in use. With temperature records, it is often clear from the range of values whether one is dealing with
1.2. ENVIRONMENTAL DATA

degrees Fahrenheit or Celsius, but as a rule unit data should be examined anyway.

GIS services available on the internet in particular often show little or no accompanying metadata, even larger commercial databases. The MetAccess commercial database ([67]), which consists of formatted Australian Bureau of Meteorology data and is much used in climate and hydrological research, does not list information on the definitions of Gale and Strong Wind. These definitions had to be obtained by email correspondence with Bureau of Meteorology sources. Happily, MetAccess does list units for all of the non-Boolean variables recorded.

1.2.12 Other Data Issues to Consider

In addition to the standard metadata issues like precision and units, we need to consider what kind of data each value represents. A single number can be a member of a discrete or continuous numerical set, represent a non-numeric class that may or may not be ordered in some way, or be a Boolean yes/no indicator. It may be directly measured or a product of an interpolation or aggregation process. Before analysis begins, we should formulate a plan to best to exploit the data, and implicit in this step is careful consideration of what data we have available.

We should also decide on a strategy for dealing with missing values, as it is rare to encounter an environmental record entirely intact. They can be treated as wildcards, or the instance can be ignored completely. The former strategy inevitably leads to loss of information, and may cause serious problems if missing values are not randomly distributed (see [4] or [85] for more on this topic). Missing value replacement methods take a number of forms. Some examples are: replacement by the value from a similar instance, replacement with the corresponding value from a geographically nearby dataset, with the mean of that variable, or with some default value such as zero. Under some circumstances, missing values are treated by replicating the entire instance for each possible value of the missing feature.

Outliers may also exist in our dataset. Variables such as daily rainfall are often noted down by hand, and errors like omitting or misplacing a decimal point can be made at this step or at the data entry stage. A rainfall value less than the precision of the recording instrument may indicate a decimal point shifted to the left, and an unusually large entry may indicate a decimal point shifted to the right or missing entirely. For suspiciously large events (these can be isolated by visualisation or by statistical examination of the number of standard deviations from the mean, for example) other climate records in the local area
CHAPTER 1. DATA AND PROBLEMS

should be checked for a corresponding large event- a cyclone in Northern Australia may have a rainfall value in the order of hundreds of millimetres in a single day, but will show up on numerous records from that region.

True outliers may also lie next to discontinuities or sudden jumps in the variable in question. For example the cyclone mentioned above will be represented in a daily record not as a single isolated extreme value, but as one or two high values surrounded by smaller but still significant rainfalls. Once outliers have been identified, they must be dealt with in some way or at least carefully noted. Where there is a simple explanation like a missing decimal point, it may be safe to adjust the value and move on. If not, and the value is certainly erroneous, they should be treated as missing points.

Another consideration is the possibility that we do not care about the difference between two states, or know a priori that the difference is negligible. Then we can simple aggregate the two values into one, and save ourselves unnecessary work. Another possibility is that under certain circumstances we know that the value of one variable is irrelevant, but not in others. Some machine learning algorithms take such 'don't care' values into account. For discussion of 'don't care' values in the context of decision tree induction, see [36]. Where the facilities exist, it is usually worthwhile to exploit them.

Variables such as those discussed above are often not normally distributed, and because of the high degree of complexity inherent in environmental systems, statistical independence between variables cannot be assumed. When using methods involving the Naïve Bayes rule, for example (see Section 2.4.8), this must be taken into account.

Choice of missing value, outlier, and other unusual data point treatment must be made with due consideration of the problem at hand, and also of data restrictions. Because the quantity of data available for an environmental analysis is often limited, it is important to get absolutely optimal performance from the data we do have. To do this, we must consider the information gleaned from metadata, theoretical performance limits imposed by uncertainty, and technical and methodological options.

1.3 Hydrological Modelling

As mentioned in the introduction to this chapter, we shall use hydrological modelling as an example sub-discipline from the broader field of environmental science. Like most model from other disciplines, hydrological models are built or calibrated from observed data. The information sequence is illustrated in Figure 1.2.
1.3. HYDROLOGICAL MODELLING

Data is collected by measuring reality, and then used to build a model summarising or characterising the information it contains. The model is then compared to the observed data, or preferably a new and independent testing dataset, to obtain some estimate of performance. However, when addressing real-world problems, the model construction and evaluation process is more properly iterative, as the model is refined and optimised.

The other important step omitted from Figure 1.2 is that of data preprocessing. This stage includes visualisation, identification and treatment of missing points, outliers, and other anomalies, estimation of uncertainties, smoothing and discretisation if required, and a host of other minor but extremely important tasks. Because data quality and quantity are often low in environmental science, preprocessing is especially fundamental to good practice modelling (see Section 1.4 and throughout subsequent chapters).

Hydrological models fall into three broad categories: Empirical, Physics-Based, and Conceptual ([156]) The empirical are obtained by direct statistical investigation of data. Data quality and quantity issues may be one of the primary reasons for this. The interdependence and distributional concerns noted in Section 1.2.12 also violate the assumptions inherent in many statistical analysis methods widely applied in other fields.

Physics-based models use the laws of physics to model the environment. An example application area where physics-based models are common is groundwater modelling, where conservation of mass and flux balances in the form of differential or difference equations are the norm. Large-scale climate models are often physics-based, but this approach has disadvantages that are exacerbated by the nature of environmental systems.

Paucity of calibration data is one of these problems. Fine temporal and spatial are often required if the resulting models are to be stable. In groundwater studies, where spatiotemporal variation can often be assumed slow, and the governing equations are relatively simple, the problem is not serious, but consider a daily catchment-scale model taking
incident rainfall as input and producing modelled streamflow. The physical laws regarding incident rainfall soaking into a patch of soil of particular type, dampness, and vegetation cover, or being evaporated, transpired, or flowing overland to stream are prohibitively complex and would require a staggering amount of input data.

Rainfall-runoff models are most often conceptual in form. This group is in some respects a compromise between physics-based and empirical modelling. Here, a rough conceptual framework is defined a priori by expert knowledge. The framework will involve a number of unfixed parameters that are calibrated for a particular situation (catchment, for rainfall-runoff models) by experimentation. For example, the IHACRES rainfall model ([43]) divides a single catchment-wide daily rainfall value into a number of parts according to parameter values and system condition at the previous timestep: losses to evapotranspiration, direct surface runoff, uptake by soil and groundwater. The entire model can be run with around five optimised parameters.

The issue of conceptual model complexity is important to the calibration process, not solely for computational reasons. Each additional parameter adds a dimension to the optimisation problem, but beyond a certain point, will not improve predictive capacity. For example, fitting a polynomial of order three with a four, five, or six degree polynomial will not result in a better fit. If noise is present in the original signal, overfitting may occur, resulting in decreased predictive power. The optimal parameter set becomes increasingly non-unique with rising number of parameters- with a large number of parameters it is possible to fit any curve at all very closely with many different combinations of parameter values. However, these values may not reflect real physical situations, and validation on an independent testing dataset will reveal poor predictive accuracy. It is therefore desirable to keep conceptual models as simple as possible while capturing the essential processes of the system needed for modelling purposes.

Although the calibration process can be computationally expensive and involve considerable manual work, the rainfall, temperature, and flow series that are involved in each iteration are often not longer than a decade. These optimization problems are usually solvable in a matter of hours of computation time, which is not considered prohibitive. The importance of good data on which to calibrate these models is clear.

1.3.1 For What Purpose?

Hydrological models are built for a number of purposes ([156]). Most models seek to find optimal water-use patterns and assess environmental impacts of human activity. We
may wish to model a catchment in such a way as we can modify the pattern of water allocation for irrigation and other uses, to investigate what the effect of those changes would be. By altering the state of the model multiple times, the optimal irrigation scheme can be investigated, keeping in mind the long-term health of the river or stream and the expected returns of farmers and other stakeholders. To take another example, the effects of building a new dam can be thoroughly investigated before the dam is put in place, to assess environmental and social impacts. Rainfall-runoff modelling is a basic tool for modelling such problems.

A related question is that of pollutant transport. Rainfall-runoff models coupled with transport modules are used to calculate the loads and concentrations of, for example, nitrates and phosphate from soil mobilisation and fertilizer applications, at given points along the river system ([91]). The concentrations of such contaminants can then be used to assess the risk of algal blooms or other environmental hazards.

Flood prediction is also performed with rainfall-runoff models, but at a finer time scale. In this case the model can be incorporated into a forecasting algorithm which updates predictions based on the quality of recent performance. The above management models are usually constructed with input data at a daily scale (which is, overwhelmingly, the scale at which climate and weather data is collected) to model changes over time periods from weeks to years. We will for the most part concern ourselves with models taking daily data as input.

Modelling for optimal management of water resources is a problem with clear and widespread relevance. Never has the importance of rainfall-runoff models been more obvious, while we see broad community debate over water management. This said, we will move on to examine the modelling process and investigating possible avenues for model improvement through the use of data mining techniques.

1.4 Further Chapters

There is a weak point in the established modelling process for environmental systems- the data. For example, rainfall and streamflow (runoff) times series, with elements numbering in the thousands, drive the hydrological modelling process, but little attention has been given to more direct methods of data analysis in this field. Unfortunately, the data are often of low quality, exhibiting an error of at least 10% in each sampled point and with a significant number of missing points where no data was recorded. This fact and the other
characteristics of environmental data such as interdependence present challenges to the data mining, knowledge discovery, and data analysis community.

In this thesis we shall explore the ways technology from these fields can be applied to environmental problems. A variety of environmental problems are explored in detail, and results presented. In addition to providing insight into the problems themselves, these investigations provide examples of how good practice data mining should be applied, and what can be accomplished with it in the environmental domain.

A review of data mining methods is given in Chapter 2. Although by no means exhaustive, a broad range of common data mining techniques and algorithms for clustering, classification, and rule extraction are discussed. In addition to these brief descriptions, selected algorithms are discussed in some detail. These examples are intended to illustrate the way data mining algorithms work, and provide some insight into the uses and history of data mining as a discipline. Popular performance evaluation methods and measures are also described, along with some preprocessing techniques. We believe that both of these disciplines are extremely important for building high quality models and obtaining the best possible results from them.

Chapter 3 contains the first application of data mining, to the identification of sites vulnerable to gully erosion. The study area is the Ben Chifley Dam catchment, which provides water to the city of Bathurst in New South Wales, Australia. Gully erosion has a strong effect on water quality in the dam, and is thus an important issue to understand and control. Classification algorithms are used, and decision trees are generated that graphically illustrate the resulting models. With visualisation and appropriate preprocessing, the investigation is largely straightforward- the value of visualisation becomes very clear as a significant barrier to analysis is identified and dealt with at this stage. This chapter serves as a first introduction to the application of data mining algorithms and to decision trees in particular.

Decision tree extractors are among the algorithms applied in Chapter 4, to the problem of discovering sub-daily rainfall information from daily climate data with classification algorithms. This is a more complex problem, where issues of goal definition, algorithm selection and evaluation, and variable selection must be and are addressed. The study uses data from six measurement stations from two very different geographical localities, so the resulting models can be compared and the possibility of regionalisation (extending models outside the original region they were developed for) explored. The analysis produces useful and interesting results, and the problem is novel.
Water quality modelling is investigated in Chapter 5, and the limitations imposed by poor quality data demonstrated. Suspended sediment concentration models were built for three nearby catchments (Jugiong, Muttama, and Tarcutta in New South Wales, Australia). First, traditional pointwise methods were applied, with very poor results. We then sought to improve the analysis by stratifying the datasets according to new variables created from the flow variable and date markers. Although the water quality variables used here are sparse and of high uncertainty, it was demonstrated that by considering the better-quality, higher resolution streamflow series accompanying the suspended sediment concentration measurements in detail, additional information could be extracted. This technique of considering data not directly related to the target variable mirrors that used in Chapter 4.

The rule-based ideas developed in Chapter 5 are extended and applied to in-stream electrical conductivity modelling in Chapter 6. A much higher quality dataset was used, with 15-minute resolution electrical conductivity and streamflow data with low noise. To avoid undue complication and processing time, both flow and conductivity variables were smoothed and the resolution reduced to a point where the number of stored points was far lower, but little information was lost. Outliers were also encountered, examined, and removed. The well-known Apriori algorithm ([3]) was modified to suit our time series data, and applied to create a model linking streamflow peaks and the behaviour of electrical conductivity.

Clustering of environmental time series is addressed in Chapter 7, along with the various uses of the process. In particular, attention is given to different time series distance/similarity measures and the effect the choice of measure has on the resulting clustering scheme. Euclidean distance with and without normalisation was used, as well as a distance measure based on information theoretic measures of series likeness. An additional compression-based similarity measure was also created using the modified Apriori algorithm of Chapter 6. This new method was tested against the Euclidean distance on a standard dataset from the UCR Time Series Data Mining Archive ([74]) and found to be good, and informative.

A dataset consisting of a few dozen streamflow records from South-East Queensland, Australia, was used throughout this chapter. Several different modes of behaviour are exemplified in this dataset, and were identified by the clustering process. The importance of selecting the appropriate distance measure for the problem at hand is noted, and some guidance towards this goal given. Problems arising from large numbers of missing values
CHAPTER 1. DATA AND PROBLEMS

were encountered, and a method developed to counter them. Also, tools were developed and demonstrated for investigation of cluster stability.

Issues arising from the general investigation of time series clustering are further explored in Chapter 8. The dataset under consideration here was far more complex. More than 50 variables relating to water quality were obtained for each of six locations in mid-Wales, United Kingdom. Chemical concentrations, streamflow, water temperature, dissolved organic carbon, and several other variables of very different characteristics were included in this extremely rich and interesting dataset, which was explored using tools developed in the previous chapter. In this way, the utility and potential of these methods was established.

Although we stress the importance of in-depth preprocessing, this thesis contains no stand-alone chapter devoted to this topic. Rather, all data mining projects undertaken discuss in detail the preprocessing work done, as an integral part of the information extraction exercise. This reflects the view we wish to express: that preprocessing is an essential step in the modelling process. In Chapter 3 in particular, it proved not merely beneficial but impossible to obtain good results without attention to preprocessing.

The topics of preprocessing, informed choice of models, application of these models and presentation of results are treated with equal attention. Chapters 4 and 7 in particular contain comparison of several classification and clustering methods respectively. Not only do we introduce multiple algorithms in this way, but the evaluation and selection of the best model or models is also demonstrated.

The analysis of Chapters 4 and 3 use only well-known, standard techniques from the field of data mining. As there are already a huge range of data mining algorithms available, this is a sensible and efficient course to use established and well-tested methods developed for other purposes. The flexibility of most data mining techniques is such that a method of analysis can be found in the existing library of data mining techniques for most purposes. Certainly, the majority of environmental scientists working using data mining will apply established algorithms, and this is as it should be.

However, in Chapters 5 and 7, concerns arising from the nature of the data at hand prompt us to develop new techniques, which are further explored and utilised on other problems in Chapters 6 and 8 respectively. The modification of existing techniques and the building of new ones is demonstrated, along with the verification and testing issues that must be considered when developing new methods. In this way we aim to illustrate the utility of data mining in the environmental modelling domain, and also to demonstrate ways in which techniques can be developed to deal with uncertain, noisy, incomplete, and
interdependent environmental data.

The new methods developed are of value for other purposes. In this thesis, techniques for water quality modelling, general time series rule extraction, time series distance quantification, replacement of missing distance values arising from the absence of matching time-periods in time series datasets, and analysis of similar and dissimilar series are presented. They add to the pool of data mining tools currently available for general use.

Lastly, significant process understanding was generated, regarding to gully erosion, rainfall intensity, electrical conductivity. The relationship between chemical concentration series was extracted for several locations. In several cases, this information is valuable and has been taken up by further studies or integrated into plans for action.
Chapter 2

Review of Data Mining Techniques

2.1 Introduction

Depending on context, data mining is defined in many different ways. We shall say that data mining is discovery of interesting, comprehensible and previously unknown rules, trends or characteristics from data. By this we mean extraction of anything that is useful and non-trivial or unexpected. A large number of data mining techniques and tools are available for the stated purpose and a selection of relevant ones are discussed in this chapter. The field of machine learning includes many of the same concepts, but may be thought of as a more general topic touching on such fields as artificial intelligence.

We will consider some of the more common methods: clustering, classification, association rule extraction, as well as issues of data preprocessing and model performance quantification. In addition, the concepts of dimensional reduction and similarity between series are discussed because of their importance in good practice data mining.

Much of the data encountered in environmental fields will be in the form of either a collection of spatial measurements (as in Chapter 3) and/or a time series (as in Chapters 5 to 8). Categorical data is also widely used in the field. Many environmental problems are attacked using databases of many variables with complex interdependencies, significant noise components, missing segments, differing statistical properties, and often several data types, for example spatial data and time series. Extracting relevant, useful, and interesting information from these datasets is a complex process where sophisticated data mining methods can be of great use.

Aside from the challenge of system characterisation using a parametric model, data mining provides many different pathways for analysis that may or may not reveal use-
ful information about the system in question. Much of the knowledge we seek with data mining techniques could be extracted by detailed and rigorous visual and statistical investigation, but the process would involve prohibitively large time and computation costs. The adaptability of data mining and machine learning techniques, coupled with the possibility for natural handling of the noisy and missing data that is so prevalent in environmental records, make them ideal for dealing with this kind of dataset.

We seek to discover this useful knowledge from raw data automatically, but as noted in [30], data mining is an exploratory process. No algorithm provides a complete knowledge of the system at hand, and no individual algorithm outperforms all others on any class of problems. Algorithms are tools that expedite the extraction of knowledge from raw data, but must be applied judiciously and after appropriate data cleaning and problem specification.

Consider for example the case of temporal data. There are two ways that we can consider such data, as time series or as unordered records of the variables collected on a given day. Using both approaches we can borrow from extensive technical bases established for other applications. For example, some tools used for stock-market analysis can be generalised for other time series, or isolated records can be considered as 'market baskets' and analysed using techniques developed for supermarket purchase analysis. A third perspective is the middle way between the two above- take one of the most fundamental characteristics in hydrology, the hydrograph. This is the quantitative response of a stream or some other medium such as an aquifer, to an input like rainfall, or its constituents such as concentration of deuterium. See Figures 5.1 to 5.3, 6.1, or Figure 2.2 (below) for examples.

When examining any stream hydrograph, we note that most streamflow and some rainfall events have a definitive length, often of a few days. It seems natural to change the granularity (or resolution) of our time series from days into peaks or events, and sometimes this is useful. Transformations of this kind can make a huge difference to the quality and interpretability of analysis, and data mining and statistical analysis can make proper treatment of data much easier for the analyst, but some human interaction with the database is still essential.

One aspect of environmental data acts in our favour. Most established data mining techniques aim to deal with anything up to millions of records, so the computational efficiency is in general very high for large datasets. In many environmental applications, the size of the dataset is comparatively small, and we can take the most appropriate method
without regard for computation time, as it is not likely to be a restrictive factor with our relatively small quantity of data (usually of the order of thousands to tens of thousands of data points at most), so more complicated statistical and data mining techniques can be considered. Of course, the small data size is not necessarily an advantage, because there is less data from which to extract information.

A huge library of data mining techniques and software packages already exists. Much of the software, for example Weka ([112]), Orange ([33]), and YALE ([94]) are available to download and use free of charge and are open source. GESCONDA ([52]), a software package intended specifically for environmental problems, is currently under development. A collection of standard and well-studied datasets for experimental and comparative purposes can be found in the University of California at Irvine Machine Learning Repository ([13]). A few, but certainly not all, of these datasets are environmental in nature.

2.2 Key Data Mining Concepts and Definitions

There are a few key data mining concepts that we will consider. They are, in the following order:

- Clustering
- Classification
- Association Rule Discovery
- Dimensional Reduction and Dominant Mode Extraction

Of these, clustering and association rule discovery are unsupervised learning methods, meaning that no target variable (see below) has been explicitly defined. The kind of trends we aim to extract using these methods are largely unknown or unspecified, whereas classification, a supervised learning task, seeks trends relating to a target or classifier variable.

Each of the above topics will be discussed subsequently in an individual section, but first, it is helpful to define some terms.

**Target variable(s)** The variable(s) that is/are the focus of investigation

**Instance** A single record from a database, also called an itemset in the literature
CHAPTER 2. REVIEW OF DATA MINING TECHNIQUES

N Total number of instances in the database
m Number of attributes in each instance

Event An instance or, if appropriate, particular group of instances connected in time
Attribute A variable present in the record
Feature Synonymous with attribute
Information Storage required to determine a given outcome or unknown characteristic from an instance
Training set Data used to build a model, rather for testing or validation purposes
Classifier An algorithm performing classification, or the class to which an instance belongs (target variable)
Cluster A group of similar points or instances within the dataset
Stopping criteria Some test determining the point at which an algorithm should end iteration

2.3 Clustering

The word 'cluster' has twofold meaning, a cluster being a group of objects, and the verb cluster, meaning to group objects according to some measure of similarity. There are many algorithms available to cluster data, mostly with the object being to produce groups such that the objects within a cluster are as similar, and the defining characteristics of each cluster as dissimilar, as possible. The number of clusters and level of acceptable dissimilarity in a cluster are considerations in this problem. We want enough clusters so that internal objects are close, but not so many that there are similar clusters. There is a trade-off between the precision of the clustering regime and understandability or usefulness of that information.

Because clustering schemes learn their own cluster or class structure rather than working with an enforced structure, this approach comes under the broad heading of unsupervised learning. This is also known as unsupervised classification, although when clustering we seek to define the classes (clusters) rather than extract the underlying characteristics of
pre-defined classes. This latter problem is referred to in the literature as classification, to which Section 2.4 is dedicated. As noted earlier, classification is an example of supervised learning. The two approaches are often combined as part of the knowledge discovery process.

In this section we will outline a clustering algorithm, and briefly examine a few of the problems associated with clustering time series specifically. The method, whose aim is to group data points into a user-defined number of clusters \( k \), is called the k-means algorithm (see [158]). k-means is much used in data mining practice, often as a benchmark against which more complex methods are compared. The algorithm runs as follows.

1. Select \( k \) different data points at random. These will be the centres of the initial clusters.
2. Assign all data points to the cluster with the closest mean value.
3. Re-calculate the cluster mean with points assigned in step 2 taken into account.
4. Repeat steps 2 and 3 until the clusters no longer change, or some other stopping criteria is met.

The k-medoids algorithm is essentially the same as the k-means, except that where we used means in steps 2-3 we now use medoids (the medoid is the m-dimensional analogue of the mode). The use of medoids is a sensible choice when the data contain outliers, because an extreme value will not skew a medoid as much as a mean value would be affected. This rough outline of an algorithm could be applied to data in many conceptual forms—isolated records, events or whole time series. However, we must decide how to measure means (or medoids) and distances between shapes, series or records. If our records contain a small number of variables, we can just use some Euclidean distance measure after appropriate normalisation of the input data. Events-based representations and time series are much more complicated to compare quantitatively. Dimensionally reduced distance and similarity measures for these types of data are discussed in Section 2.6.

A simple clustering example is illustrated in Figure 2.1. This example is taken from Chapter 7. Here, we have the geographical locations of 21 streamflow gauges in Universal Transverse Mercator units. The series are clustered using the k-means algorithm with a straightforward Euclidean distance measure between geographical locations. In the figure, locations are coloured by cluster. Lowering the number of clusters from five to four would likely result in the aggregation of the red and green clusters. If \( k \) was set to three, the
yellow and purple clusters would probably also become one. In practice, $k$ is usually set by expert knowledge or trial and error.

When computing clusters based on the flow series themselves, choice of an appropriate distance measure becomes less obvious. Figure 2.2 is a plot of all 21 flow series over a short time period, and illustrates the problem of clustering time series. Chapter 7 discusses time series clustering in detail.

Some other methods for producing clustering schemes take splitting or agglomerating hierarchial approaches, or build clusters with pre-defined characteristics (ie distributions). The latter is known as mixture decomposition (see [150] page 121). An example of the output of an agglomerative hierarchical clustering algorithm, is given in Section 7.6.

### 2.4 Classification

Classification is used to extract a relationship between particular instances or features in an instance and the value of some unknown categorical target attribute or label, called the classifier. The classification may be in the form of a decision tree, posterior probability estimates, logical rules, or some other type. In some circumstances classification is referred to as 'the discrimination problem' (eg in [35]), because the aim is to create a model that discriminates between instances of differing class. In some cases the classifier takes one
2.4. CLASSIFICATION

Figure 2.2: Streamflow Series for South-East Queensland Gauges, Coloured by Number of two values (the classifier is binary), and in others more than two classes are defined (multiclass problems).

2.4.1 An Example

The nature and usefulness of classification is perhaps best illustrated with an example. In a daily rainfall dataset, often no information is available about rainfall distribution during the day, for assessment of the intensity of a rainfall event. We do not know if the rainfall measured on a given day fell evenly over the whole 24 hour period or in one short intense burst. In the context of rainfall-runoff modelling applications, intensity information would be extremely useful, as the nature of the streamflow response varies with rainfall intensity. The intensity of a rainfall event not only alters the flow peak steepness and timing, but is also an important determining factor in the mobilisation and transport of solutes and suspended particles.

High-resolution rainfall data, by which we mean data collected at intervals of order less than a day, is expensive and difficult to collect and much rarer than daily data. Therefore, we would like to find a way of determining the approximate intensity of a rainfall event from daily data. This seems like a very difficult task, but other daily climate data may be available to assist: maximum and minimum temperatures, relative humidities, solar radiation, sunshine hours, evaporation, and other variables are often measured daily.
Using the daily data series from all the available climate variables, one approach would be to hypothesise as to where in the rainfall series intense events occur. Using pre-existing physical knowledge of the catchment or catchment type, it may be possible to find, for example, an approximate set of physical characteristics for a summer storm day with temperature, humidity and rainfall above set thresholds, a date within the likely bounds, and so on. This is clearly a very vague approach that is unlikely to be workable in most areas.

Instead, we select all those sites where high-resolution rainfall data does exist for at least over some temporal interval. Then we can apply some simple criteria to the high-resolution data, for example a minimum amount of rain must fall in a certain small time interval on a given day, for that fall to be flagged as an intense event. In other words, if \( R > R_{thres} \) and \( \delta T < T_{thres} \) then Event = Intense, else Event = Not Intense.

This new variable forms the classifier for training purposes, and we can now use a data mining algorithm to automatically extract those combinations of daily climate data characteristics which tend to occur on a day with a value of unity in the intensity attribute. In the process we will generate a decision tree or other model for automatically deciding whether or not a rainfall event is intense, which can then be applied to daily data where higher resolution data is not available. The Boolean intensity yes/no marker is called the classifier and the process of building and applying the model, classification. When this process is actually applied to real data useful information can be extracted. In [142] and [146] the rainfall intensity decision tree model building process is illustrated and results presented.

Of course, it is in theory possible to build a classifier (especially a decision tree- see Sections 2.4.2 and 2.4.8) manually, by examining system data or using expert knowledge. For example, research detailed in [154] indicates that with adequate visualisation novice users can build good decision trees up to a certain level of complexity, but only where relationships between the target (classifier) variable and the descriptory features can be viewed in two dimensions (ie where at most two descriptory variables are involved in each relation). However, manual tree-building is expensive in time and labour hours. Also, a limit of around 10 relevant variables for this kind of building scheme comes into play. A few schemes have provision for user interaction with the tree building process, but the practice is not common. Most recent work focusses on obtaining a high degree of automation, so that large and complex databases are more readily amenable to analysis.
2.4. CLASSIFICATION

2.4.2 C4.5 Decision Trees

We will examine in detail the mechanics of one popular classification algorithm, and con­sider briefly a handful of alternative approaches. The algorithm of focus will be Quinlan's C4.5 [117], which is very well known and used in the data mining community. For the sake of simplicity we will assume all instances belong to one of two classes (0 or 1) as above, although generalisation to more than two classes is straightforward.

The decision tree is built up layer by layer, splitting one variable to create two branches. Eventually, this results in a tree like the one shown in Figure 2.3. The approach to creating classification models discussed above is called a top-down or general-to-specific approach because the strategy is directed towards making rough rules or sub-models covering all the dataset to more specific branches covering smaller regions. The opposite approach, bottom-up or specific-to-general, starts with very narrow sub-models covering only a subset of the data and successively broadens these rules until they cover enough instances to be useful. An example of this strategy is the AQ family of rule extraction algorithms (see Section 2.4.8).

Decision tree methods are easily interpretable to non-experts and may yield a great deal of system information. They are in general relatively computationally efficient to extract. For these reasons and historical ones, decision trees are perhaps the most popular classification schemes in use today. However, certain limitations on this classic decision tree structure exist. The most important principle to keep in mind is that the area in attribute space covered by each leaf of the tree is a hyper-rectangle with surfaces parallel to the axes. That is, the region covered by the leaf is necessarily bounded by flat planes parallel to those axes. The bounding planes may be defined by relations such as $x = 2$ or $x = 56.9$, but not $x = 2.5y$ or $x = y + 7$. There are models which allow for boundaries not of the former type, but C4.5 is not one of these.

The variable on which the split is performed is located by examining all possible branching schemes and selecting the one with the greatest 'purity' in the generated nodes. Simply put, a pure node has only one value of the target attribute, where a less pure node contains a mixture of values. There are many methods for quantifying purity. The C4.5 algorithm typically uses the maximum information gained (see Equations 2.2 and 2.3 below) by making the split, and the popular commercial algorithm CART uses the Gini index discussed in [149] and [44]:
\[ Gini(c_i) = 1 - \sum_j p_j^2 \] (2.1)

Here \( p_j \) represents the fraction of instances in the relevant node that takes the \( j \)th value. However, choice of purity function has less impact on the output decision tree (within appropriate limits) than choice of pruning scheme- the way a tree is reduced in sized when overgrown and overfitted. See Section 2.4.4 for a brief discussion on pruning techniques.

Note that this process of splitting on node purity, which looks one step ahead (called 'one-level lookahead' in some literature, eg [44]), would be ideal if the contributions of each variable to the classifier value were linearly separable. The iterations splitting the data continues until all instances on every branch of the tree are covered by a rule, or some stopping criteria like a minimum accuracy per node is reached.

Below in Figure 2.3 an example decision tree is given. Here, the object is to classify a given year as a leap or non-leap year. The tree splits on the single variable \( \text{year} \) number twice, although in practice two additional variables have been created- \( \text{Year} \mod 4 \) and \( \text{Year} \mod 100 \), and each of these split once. This tree has a total of five nodes, with two decision nodes (branching points) and three class assignment nodes (leaves). Red lines denote a failure of the condition at a decision node, and green lines the condition holding.

This tree classifies most years correctly. In fact, the only cases where it fails are years exactly divisible by 400. Where this condition is true, we know that the \( \text{Year} \mod 100 = 0 \) decision is over-ridden and the year is in fact a leap year. Obviously, very few years (one in 400, of course) are divisible by 1000 with no remainder, and if the tree in Figure 2.3 was trained from a dataset without any of these years, any error estimates generated using only training data would appear perfect. If a testing set containing millennial years was introduced, then a lack of accuracy would be noted. But even in a very large database of years, it is not likely that there will be many of these points, and thus the tree below will be judged favourably.

If a small number of millennial years exists, making up a tiny proportion of the training set, it is possible that the tree building algorithm will still settle on the model in Figure 2.3, because the false classification rate would still be very small in comparison to the majority of correctly assigned cases. Up to this point we have also assumed that the training data is completely free of noise, which is almost never the case. If a few leap years in the training set were incorrectly labelled as non-leap and/or some non-leap years were labelled leap, we
2.4. **CLASSIFICATION**

would not want the tree extraction algorithm to be too sensitive to aberrant behaviour in a small number of instances. This is especially important for environmental data which is inherently noisy. C4.5 does not require absolutely pure nodes at any stage, and the level of noise tolerance can be adjusted by varying input parameters.

As we noted earlier, by 'purity of node' we mean a quantitative measure of how close each side of the split is to containing only zeros or only ones. In the whole tree, each node at the end of a branch or leaf will ideally contain (in the binary case) only ones or only zeros, making it completely pure, giving unambiguous decisions, e.g., intense rainfall or not intense rainfall. The purest nodes give the most information, and we would like our nodes to reach complete purity as quickly as possible so as to generate the smallest final decision tree, although using purity as a splitting criteria does not necessarily minimize the tree size or find the optimal tree.

Another choice is the information function as given by [158]:

\[ I(p_0, p_1) = -p_{0,j} \log_2(p_{0,j}) - p_{1,j} \log_2(p_{1,j}) \]  
\[ \text{info(split)} = m_1 \cdot I(p_{0,1}, p_{1,1}) + m_2 \cdot I(p_{0,2}, p_{1,2}) + \ldots + m_k \cdot I(p_{0,k}, p_{1,k}) \]  

The \( p_{i,j} \) are proportions of the classifier valued \( i \) in a given branch \( j \) after a split. The \( m_j \) are the fraction of all instances in that arm of the branch from the total. This is the only functional form that satisfies certain necessary properties discussed for example in [158], but as noted in [57], these measures are biased towards features with a greater number of possible values (modifications can be introduced to counter this bias, but we
will ignore these for the present). The information over a branch must be equivalent to all the component information, and the information at any node must be at a maximum when the number of instances of each classifier are equal. When the node is pure the information is zero. For a very readable discussion of information measures in the context of decision tree construction, see [153].

We choose the branching scheme where information gain (information at the level above the splitting node minus information gained by this branching) is greatest. The variable on which we have branched may then be discarded from information gain calculations at the next level. We would also like to limit the complexity of the final tree, so splitting over a variable with many possible values should be avoided if practical. A modification of information gain, called gain ratio, where the information function is evaluated as if every classifier were the identity, is sometimes used. Then we are measuring the number of branches resulting from a split and counting the instances in each partition. Information gain is divided by this measure to get the gain ratio (see for example [158], [119]) for ranking possible branching schemes.

Additional stopping criteria may also be applied if appropriate to the problem at hand. For example, we may decide to limit the number of final nodes (leaves on the tree). Usually, the trees output from C4.5 and other tree algorithms are much larger in number of nodes than they would ideally be, so they need to be pruned back to an appropriate size as discussed in Section 2.4.4. After applying some stopping criteria, we will have a tree like those shown in Figure 2.3 and 2.4. At each terminating (leaf) node, the subset of instances covered will be close to purity. The obvious step is then to assign the leaf to the most common class, but as noted in [155] we must take the class distributions (the number of occurrences of each value of the target variable) into account if the training and testing datasets do not share the same distribution. Also, we may have to take unequal misclassification costs for the different classes into account (see Section 2.4.9).

This basic divide-and-conquer approach needs a few modifications before it can be applied to variables in real numerical form. Branching on less than/greater than criteria solves this problem. Finding the best point at which to split the data requires a little computation, but not so much that it cannot usually be accomplished by directly testing possibilities. The alternative is to discretize the input data before applying the tree-building algorithm (see Section 2.6.1). This was undertaken in the following example ([40]), with some real variables. This study aimed to create a model for the wet and dry season crop choices of small landholders in three catchments in Northern Thailand. The dataset
2.4. CLASSIFICATION

Figure 2.4: Wet Season Crop Decision Tree

contained 471 instances of more than 20 variables each, recorded by extensive survey of households. Variables under consideration included plot area and type, available water, cash, and labour, presence of livestock, frequency of drought and flood, and off-farm income sources. Figure 2.4 illustrates the output tree for the wet season.

Although more complicated than the example of Figure 2.3, the wet season crop choice tree follows the same format. At each branching point, the data is split into two parts. Here, the left branch carries the instances for which the splitting decision is true and the right those for which it is false. This tree was generated using the whole dataset, but performance was evaluated with 10-fold cross-validation (see Section 2.4.9) to ensure that it is a good representation of the system. The performance of this model will be discussed in detail in Section 2.4.9, although it is worth stating here that it was found to be excellent. Some pre-processing of the database was preformed so that detection of smaller classes was not overwhelmed by the majority crops of rice and longan, as discussed in Section 2.4.6.

2.4.3 A Consideration: Data Types, Missing Values, Noise

We now have an idea of how a classification tree is built. Before proceeding further we should consider the data types and quality usually encountered in the environmental sciences. This was discussed at length in Chapter 1, but now we must consider the way data mining algorithms deal with data of varying characteristics. To apply the C4.5 algorithm
detailed in Section 2.4.2 the problem at hand must have categorical class values. However, we may have a discrete or continuous numeric classifier. For the discrete case, we can simply ignore the ordering of class values and treat the classifier as categorical, and for a continuous classifier we could discretise (see Section 2.6.1 for a brief discussion) and treat the new values similarly, or use less than/greater than criteria in some form.

Alternately, another classification method could be chosen. For example, a kernel method classification algorithm, instance-based methods, or any variety of regression (see Section 2.4.8) predict continuous values quite naturally. Another possibility is the hybrid approach called a model tree. Here, the data is split into a tree in a fashion analogous to the process described in Section 2.4.2, but at each leaf a regression model is created.

Environmental records often contain missing points for one reason or another. Clearly, the simplest way of dealing with a missing attribute in an instance is to discard that instance, but removing all such instances is wasteful of information that may already be sparse. Therefore, we should look to more sophisticated approaches if possible. The missing value can be replaced by the attribute mean or mode, or set to equal the corresponding attribute value of a whole instance otherwise similar to the incomplete one. As detailed in [44], the CN2 algorithm fractionally splits the instance into two copies so that it may satisfy both sides of a split on the missing variable.

Dealing with 'don't care' values, where the value of an attribute is irrelevant, is simpler. Following the lead of CN2, we may simply replicate the instance and assign one or more replicate to each side of the split. Noise, however, is difficult to deal with. The most appropriate strategy is to be fully aware of the quality of our data set and the potential for overfitting or fitting to errors. In Section 2.4.7, we discuss measures of the base level of information that can be simply extracted from a particular dataset so that subsequent performances can be evaluated in context. Calculation of theoretical error bounds can also be a useful guide to appropriate performance levels—when predicting a variable with error of, say, ±10% at each point to an apparent higher accuracy of 3%, say, overfitting is obviously occurring. See Section 2.4.4 for more on this topic.

2.4.4 Pruning and Tree Sizes

Checking for and if necessary removing overfitting should be a part of the analysis process for any model. Once we have a complete model, we may wish to 'prune' the tree, consolidating or removing branches with little informative value or covering few instances. In fact, some in the data mining field believe that pruning methods are more important than the
choice of tree building algorithm, because several approaches may return the same tree or similar trees, but pruning methods widely differ. A comparison between several examples can be found in [96].

Pruning results in a smaller tree with some loss of training accuracy, but if pruning is judicious, overfitting should be reduced and the training errors resulting from the pruned tree will be closer to the error that will be incurred when applying the model to further, unclassified data. We must also take into account that most data requiring classification will have some noise in the training and testing sets. Therefore, we should not fit a model to correctly classify every instance, because some of the branches will be misled by small amounts of erroneous data.

An overly large, poorly pruned tree that overfits the training data will produce a significant drop in performance from training to testing. A cross-validation procedure (see Section 2.4.9) or similar test of stability is also recommended to test how well pruned a tree is. The structure of the tree will ideally remain constant or near constant across cross-validation runs. Those sub-branches that fluctuate are ideal candidates for pruning.

Whittling down the size of the tree also makes the representation more compact and easier for humans to understand. Various pruning methods are discussed briefly in [44] for the C4.5 and CART ([149]) packages. Automatic pruning methods are available in many data mining software packages. The problem of selecting which branches to be pruned is equivalent to selecting the most relevant or dominant branches (rules) in the tree, and will be dealt with for association rules in Section 2.5.4. As covered in some detail in [44], a cost function can be defined using both the size of the tree and error rate or rates to choose a tree with an optimal balance between size and performance. As noted in Section 2.5.4, similar techniques can be used for pruning trees and selecting rules, and pruning for bagged and boosted trees (see Section 2.4.5) and [160] among others.

2.4.5 Attribute Selection, Bagging, and Boosting

A number of techniques exist, distinct from the true decision-tree building algorithm as exemplified in Section 2.4.2, that can be used to enhance the performance or interpretability of a classifier. Interpretability of a model should not be undervalued as a virtue- it is one of the greatest advantages of using a decision tree approach over, for example, a Bayesian classification model. However, when a large number of variables exist in the training dataset, output decision trees may be expansive and complex, and hence difficult to understand. One way to prevent this is to be selective with the input variables. It
may be possible to eliminate dependent variables or those without much bearing on the classifier value.

Principal Component Analysis (PCA) (see for example [150]) may be of use here. PCA uses eigenvector calculation to isolate those attributes that contribute most to the variance of the dataset (or some sample variance), and summarises the instance vector using the first few of these components. Note that pre-processing in this way is appropriate for numerical attributes only, and some normalisation of attribute may be called for. Principal component analysis was used in [71], [114] (both in the environmental field) and numerous others. The problem of outlier detection in environmental data was also explored in the former. Other dimension reduction techniques are discussed later in Section 2.6.

Several techniques exist specifically for attribute selection in data mining, built around one of two concepts: searching through the space of possible combinations of attributes and evaluating the performance of some learner, or examining the statistical properties of the feature vectors and using some criteria to reject a subset. The former are called wrappers and the latter filters ([57]).

A simple example of a wrapper is the sequential search where attributes are either removed or added to the set of variables in use (often initially defined as all or none of the attribute set), the resulting model evaluated in some way (see Section 2.4.9), and the candidate group resulting in the best model becoming the new set of selected attributes and so on, until a stopping condition on number of variables or model performance is met. The genetic algorithm (GA) approach to this problem is also popular, partially due to the simple formation of the problem in GA terms. Considering each possible combination of attributes as a binary vector B such that \( B_i = 1 \) if attribute \( i \) is present and \( B_i = 0 \) if attribute \( i \) is absent from the combination, the problem is perfectly posed for a generic genetic algorithm to solve by 'evolving' likely binary vectors. An example of this method can be found in [34], for benthic macroinvertebrate data. [152] gives a more detailed look at the virtues of genetic algorithm attribute selectors, and for a survey of common attribute selection techniques, see [97].

Of course, with small numbers of attributes expert knowledge can be used to select which attributes are most relevant, or an analysis of dependencies between variables, for example using cross-correlation techniques or multiple runs of learning algorithms as described in [63], may also reveal important information but is expensive in time and manual effort.

Bagging (Bootstrap AGGregatING) ([14]) and boosting are two of the many methods
for aggregating slightly different decision trees into a single model. The super-classifier is called an ensemble or meta-classifier or -learner. It combines the predictions of each of the submodels into one single set of predicted values. For example, if the models $A$ and $B$ predict that the value of the target at a particular instance should be $t_1$, but models $B$, $C$, and $D$ predict $t_2$, a simple meta-learner might select the value $t_2$, which holds the majority of predictions.

Significant increases in accuracy can often be achieved by a bagged or boosted model over a simple tree. A loss of model performance can occur, but recent work (as cited in [119]) has provided fairly strong rules guaranteeing performance increases under certain conditions. In both methods, multiple trees are generated using variations on the training data, and then combined by some voting process into a single model.

Bagging selects from the $N$-instance training set (with replacement) $N$ instances, so that the new training set is the same size as the original, but some entries may appear multiple times and some not at all. The classifications generated by the aggregate model are simply the classifier given by a majority of component trees. Ties are resolved arbitrarily. An extension to this method, known as arc ing (Adaptive Resampling and CombinING) ([15]), has been developed using resampling according to some distribution extracted from the errors of the last model. Better results can be achieved this way, although it has the disadvantage of additional complexity.

Boosting uses weighted instances to generate decision trees. The first tree is computed with every instance weighted at $\frac{1}{N}$. The weights are then adjusted according to the errors arising from this tree and a new tree generated. After $T$ trials the trees are combined by comparing votes for a given classifier in a manner similar to the combination procedure for bagging. The votes under a boosting scheme are, however, weighted using previously generated errors before comparison ([129]).

To improve the predictive capacity of a classification scheme, it is sometimes appropriate to combine the models generated by two or more different algorithms in another way. Where the accuracy of an algorithm is highly dependent on the input features, ie the model is accurate for some types of input but not for others, we may use this method of combination to build our meta learner. Beginning by classifying in some fashion where each algorithm does and does not classify well, we can then use this information to decide which scheme to use for each particular instance to be classified on an individual basis.

Methods combining multiple classifiers, such as bagging, boosting, and other meta-learning schemes, may be considered as state-of-the-art. As noted in [44]
A tree that after pruning still remains too big to be comprehensible is a sign that a more powerful description language is required.

The meta-learning tactic is the 'more powerful description language' that has been adopted by the machine learning community.

However, no method is perfect for all problems. Meta-learned models lack the easy comprehensibility and clear visualisation that simple decision trees and model trees boast. The classic C4.5 scheme discussed in Section 2.4.2 is still oft used as a benchmark for new algorithms and techniques, for example in [154], [113], [63], [115], and [57].

### 2.4.6 Class Distribution and Resampling

It is easy to assume that the natural distribution of a dataset is the ideal distribution to capture in the training set. However, this is not always the case— the detailed study [155] finds that the optimal distribution is often not the natural. Most algorithms, including C4.5 (Section 2.4.2), tend to predict the majority classifier (the class holding the most instances). For example, if 90% of our classifier values are unity and 10% zero, a classifier could simply assign all entries to unity and still have 90% accuracy (see Section 2.4.7) in training and validation. The prior probability of any given instance is weighted towards the majority class. Also, minority classes tend be more difficult to learn because less examples exist for training. Thus the information about class characteristics and boundaries is less easy to extract, and new minority class instances are more likely to be different to those in the training set.

Two points arise from this fact: first, a simple percentage accuracy is often insufficient to capture true model fit to data; and second, we must take careful note of class distributions. Other measures of classifier performance are addressed in Section 2.4.9.

Resampling a dataset can be performed in a number of ways. If the training set size is restricted only by computational expediency, instances can merely be selected from the larger dataset to form a training set with the desired distributional properties, but in the environmental sciences there is rarely a surfeit of data. We can simply delete instances from the training set until the chosen distribution is reached, or we can replicate or create new instances by modifying or combining the old. Deletion has the obvious disadvantage of throwing away precious data, but multiplication also has its downside.

Using a dataset resampled by multiplication, care must be taken that the covering sub-model (for example rule or leaf of a decision tree) does not cover only one replicated
instance. The node in this case is clearly overfitted, but the illusion of multiple instance coverage may hide the fact. Thresholds on the number of instances covered by each leaf must be carefully enforced, and we must also be aware that for every single instance in the original dataset, those resampled by multiplication will contribute more strongly to the error calculation for each class, counting once for every replicate as well as for the original. Note that majority class performance may also suffer under resampled conditions because the bias towards assigning instances to this class will be reduced.

These problems may be ameliorated by combining two or more minority-class instances to form a new, different minority class instance, but to do this while preserving the characteristics of the system is a difficult task in itself. Where two or more minority classes exist, it may be useful to merge these two into a new, larger class, but the viability of this approach is very much dependent on the system at hand. Any ability to differentiate between the aggregated classes is of course lost, and the approach may be counter-productive if such classes are very different in character.

In the study [40], first discussed in Section 2.4.2 above, combining small classes worked well. There, the object was to classify crop type for small farmers in a district of Thailand. With the original class structure, two overwhelmingly large classes (Rice and Longan) dominated the classification, with many minority class instances (Banana, Onion, Chilli, Basil, Tomato) largely allocated to one of the two massive classes. Combining the smaller classes into categories such as Fruit Trees, Herbs, and Vegetables gave enough weight to the new classes to produce a meaningful (and very accurate) model. Performance in some small classes was still low, although much improved for others. Attention then turned to resampling by multiplication. For details on that process, see [40] or [41].

Another approach is to alter the classification scheme, or adopt a cost-sensitive classification algorithm. Here, we may wish to define the cost of a false positive and a false negative unequally (note that a natural generalisation to multi-class problems exists). Where the training set is known to have a different distribution than the test set, some accounting is recommended. For example the re-assigning of C4.5 leaves as described in [155] or targeted pruning techniques.

While the optimal distribution for learning remains unknown a priori, [155] states that using an even distribution usually improves classification over use of highly skewed distribution, even if the latter is the 'true' distribution. As with choice of distance measures, the goal and nature of the problem must be taken into account when considering resampling methodologies. The class distribution should always be examined as a pre-processing step.
2.4.7 Benchmarking

Data mining is an explorative process directed by human interaction, although the individual algorithms perform autonomously. At each stage, it is useful to know how much information we can reasonably expect to extract from a given dataset in the context of our particular problem. In other words, how well a learning scheme is performing. Statistical measures or Receiver Operating Characteristic curve analysis (see Section 2.4.9) can be used to quantify the performance of an individual scheme, but some problems are intrinsically more difficult than others.

Benchmarks can give us some idea of the minimum information that can easily be extracted from a dataset, so we can have a point of reference for the performance of other learners. Most simply, we can take the classification scheme that assigns every instance the majority class, or the attribute mean if the target class is numeric. This gives us a very simple criterion for an acceptable model: if performance is no better than that of the simplest classifier, it should be discarded. Recall the problem discussed in the previous section, where the target class contained 90% unity values and 10% zero values. Any classification model with an accuracy of less than 90% when applied to this problem is probably useless.

A slightly more sophisticated benchmark classification learner is OneR (see [66] or [30]), which uses a single feature to explain as much of the target behaviour as it can with multiple splits. In addition to its value as a benchmark, OneR can also be used to compare the individual contributions made by a number of single attributes in an intuitive way. Often, the correct classification rate of this algorithm is surprisingly high ([66]).

2.4.8 Other Algorithms

There are many other classification algorithms based on entirely different concepts, divide-and-conquer being perhaps the simplest. Bayesian methods, neural network and genetic algorithm forms are among the conceptual bases of some other approaches. We will briefly review some of the more popular supervised learner families.

Naïve Bayes The Naïve Bayes algorithm (see for example [164]) and its variants classify individual instances by estimating posterior probabilities with Bayes' Theorem. It can be used to predict numerical classifier values. Or, with a simple transformation of an $n$-categorical classifier into $n$ binary categorical classifiers (such that the value is unity in the $n^{th}$ classifier where the instance belongs to the $n^{th}$ class), it can be used to
predict categorical classifier values with multiple classes. For example, if the target variable takes one of \( n = 3 \) values, \( t_1 \), \( t_2 \), or \( t_3 \), the target could be transformed into three binary variables of the same size. The first target would be filled by placing a one in all positions corresponding to \( t_1 = true \) in the original target and zero everywhere else. The second and third binary target are created similarly with the criteria \( t_2 = true \) and \( t_3 = true \) respectively. Of course, the \( n^{th} \) binary target is uniquely defined by the others, because if a given point is unity in one of the other binary targets, it must be zero in the \( n^{th} \), and if it is zero in all other targets, it must be the \( n^{th} \) value and hence will be unity. Note that this method can also be used in other circumstances as well.

In a two-class system, the class assigned to a data point is given by:

\[
v = \arg\max P(v_j | a)
\]  

(2.4)

Above, the symbol \( v_j \) denotes either class 0 or 1 and \( a \) is the vector of attributes for the instance in question. The above equation is obtained directly from consideration of Bayes’ theorem:

\[
P(v_j | a_i) = \frac{P(a_i | v_j) \times P(v_j)}{P(a_i)}
\]  

(2.5)

Note that \( P(a_i) \) is constant for all \( v_j \). \( P(v_i) \) can be estimated by the fraction of instances where \( v_i \) is given as the classifier. Thus, because we are searching for maximum probabilities over a single value of \( j \), we may eliminate these two terms from Equation 2.5. Now, if we apply the Naïve assumption of independence between attributes, ie that

\[
P(v_j | a) = P(v_j) \prod_i P(a_i | v_j)
\]  

(2.6)

then this can be substituted into Equation 2.4 to obtain

\[
v = \arg\max P(v_j) \prod_i P(a_i | v_j)
\]  

(2.7)

which we can compute ([164], [58]).

The Naïve assumption is often false, and certainly this is the case for most environmental data, even with careful attribute selection. However, it is now known from a variety of experiments and theoretical work (see [47] for an example) that simple implementations of the Naïve Bayes approach may yield good classification models, even with data that flagrantly violates the Naïve Bayes assumption. This is especially true for categori-
cal classification, as demonstrated in [47]. Where this is not the case, we must abandon Equation 2.6 and use more complex probability estimation methods, either based around some estimation of a particular parametric distribution (eg the gamma distribution) or one of the non-parametric methods described below. It is also known (see [30]) that the presence of non-informative attributes in the dataset can adversely affect the Naïve Bayes classification scheme.

It can be argued (as in [61]) that the Naïve Bayes method is a statistical approach rather than a machine learning or data mining algorithm, but as it falls under the definition of data mining made in the Introduction, it warrants inclusion here.

**Non-parametric and Kernel Methods for Density Estimation** An extension of the Naïve Bayes approach is the probability density estimation family of classification algorithms. They seek to classify by the greatest posterior probability, but do not necessarily place conditions on distributional properties of the dataset. They can also be used for clustering, by directly estimating the density of the data in the appropriate space. Methods where no prior distribution is assumed to hold are called non-parametric ([131]), and methods where the distribution is known or assumed are called, accordingly, parametric. For example, it may be assumed that the distribution in question is a mixture of normal distributions. Probability functions for a given attribute are characterised by an estimate of density function, from which the probability function can be obtained by integration.

Density can be modelled by a weighted and smoothed sum of some functions $K(\cdot)$ on the data, called kernel functions. These form a basis for representing the data in much the same way that, for example, Fourier frequency spectra can be used to capture the behaviour of a series. Happily, differentiability and some other useful properties that can be built into the $K(\cdot)$ carry over to the density estimate ([86]). With appropriate choice of kernel, the transformation can also map the problem to a lower dimensional space, to ease computational burden of further analysis.

**Instance Based Learners** These are otherwise known as case based, memory based, or nearest neighbour classifiers. Very simply, an instance based learner treats every m-feature instance as a point in m-dimensional space, and allocates target attribute values to unclassified instances according to the nearest classified points ([61]). Note the parallels between this idea and that of clustering (see Section 2.3). For a discussion of distance measures for high-dimensional problems, see Section 2.6. For purely numerical attributes, a Euclidean distance or weighted sum may suffice as the distance measure. Note that the mean and distributional properties of each variable must be taken into account, and
features containing little information may adversely affect the quality of classification. Similarity measures are discussed in the context of case based reasoning in [109].

Most practical applications of instance based learning schemes are not quite this simple, as they select from instances that are close in different ways. Storing all the instances (points) takes up valuable memory space, and their retrieval and comparison takes time and cache or register space, and classified instances containing noise in one or more attributes can skew classification. The selection of instances, a discussion of which is contained in [16], is a non-trivial problem. A variation on the instance based theme are so called elite nearest neighbour schemes (see for example [30]), where only a selection of features contribute to the distance calculation.

Note that nearest-neighbour methods do not produce a model in the usual sense. This is a limitation for two reasons: firstly, no succinct characterisation of the system is obtained for drawing physical conclusions, and the estimation of classifier values for new data requires storage of all the training data or a large subset of it.

**Model Trees** This class of classification scheme is similar in appearance to decision trees, except at each leaf we have a small regression model that outputs a continuous value. As noted in [47], the model tree builder M5' (pronounced em five prime) is known to be a high-performance algorithm in terms of accuracy, and it also has the decision tree style advantage of easy visualisation and interpretability of model.

**Neural Networks** An overview of this family of methods can be found in [128]. The reputation of neural network models as 'black boxes' has no doubt detracted from the use of this method, but simple feedforward neural networks can yield useful information about the dependency of the classifier on attribute values. Good visualisation is essential for clarity in a system as complex as a neural network, to view the web of linear relationships between attributes, temporary variables (hidden nodes), and classifiers built by a neural-network search scheme. Stopping criteria must be applied to neural network learners with care, because they may converge to local minima or not at all.

However, when feedback is introduced, or the number of layers of hidden nodes is increased above one, the web becomes much more complex and harder to manually interpret and the 'black box' label becomes more justified. A brief overview of neural nets as Principal Component Analysis, an approach providing much information about the system (see Section 2.4.5) can be found in [130]. Neural network computational complexity also rises rapidly when the number of attributes is more than a handful, and so it may become a prohibitive factor. Note that as with regression methods, attributes must be numerical
or mapped to numerical (except for a specialised class of network).

Neural nets are one of the few machine learning techniques to find widespread use in the environmental sciences, perhaps because of the popular discussion of the philosophy behind them. Other less complicated data mining methods can be applied to similar problems with as much probability of success and greater interpretability.

**Alternating Decision Trees** This method was originally introduced by Freund and Mason in 1999 for two-class problems, although several methods for generalisation to multiclass problems were later devised (see [65] for discussion of various wrapper methods). The motivation behind alternating decision trees is that boosted (Section 2.4.5) classifiers, while often very accurate, are often too complex for easy human interpretation. For algorithmic details, see [49].

In addition to an interpretable decision tree, the alternating decision tree algorithm (ADtree) described in [49] produces as a by-product a set of classification margins, which provide an estimate of confidence at each leaf of the tree. The classification margin is expressed as a real scalar value in the range $-\infty$ to $+\infty$. A large (in practice something like $\leq -1$) negative value implies high confidence in classification as the negative (or null, or false) case, and large (say $\geq +1$) confidence in classification as the positive (or true) case. The above paper also notes that decisions with classification margins close to zero can be removed from the tree to give a partial classification scheme with greater confidence, not covering those instances that prove difficult to classify.

**User-defined Classification Models** While manual construction of classification models is in general very time consuming, there are cases where it may be appropriate. The form of the model is usually a decision tree or some other visually-oriented type. In [154], it is found that with good visualisation tools non-expert users can create decision trees of a quality comparable to those returned by the standard C4.5 algorithm on certain benchmark datasets taken from [13]. While some predictive power may be sacrificed, the manually constructed trees of [154] are typically smaller than automatically generated trees built from the same dataset.

However, the study also found that the number of attributes in the dataset limits the ability of humans to construct competitive decision trees- if the number of attributes is more than a handful, C4.5 creates better trees. Of course, generating trees with C4.5 is very quick even for large datasets, while the manual construction method may take hours of intensive work for a single tree.

Human interaction with the model-building process has the advantage that expert
knowledge may be used and fuzzy or subjective data can be incorporated easily (although repeatability may be an issue). Examples of model building with this kind of qualitative approach can be found in [56], [99], and [20]. Under some circumstances, such as where strong expert knowledge exists and the number of features is small, it may be advantageous for the user to manually build a decision tree or other classification model. Possibly, this approach could be combined with a learning algorithm such as C4.5 to build trees with certain expertly-defined constraints on form or hierarchy.

Specific-to-General This is a rule creation strategy such as used in the AQ ([161]) and CN2 ([44]) algorithms. It begins with rules that need not cover more than a single instance and iteratively relax the requirements until the rule is as general as it can be while still covering only instances of one class. Some noise tolerance can be built in at this point. Note that this approach will lead to a classification scheme that may not cover all training instances, and may be applied in an unsupervised fashion (without a specific target) to produce a summary of strong trends in the database.

Support Vector Machines As mentioned in Section 2.4.2, most decision trees use hyperplanes parallel to attribute axes to partition the data space into regions of different class. Support vector machines (see for example [73], or [55] for an environmental application) seek an optimally partitioning hyperplane with no such constraints by maximising the empty margin around that boundary. In the case where classes are not linearly separable (ie attributes act together, and contributions cannot be separated), the method can be extended to operate in a feature space formed by applying some non-linear kernel transformation (see above) to the data. The method assumes a two class problem, so additional processing is required for multiclass problems.

2.4.9 Evaluating Classifier Performance, and Cost Matrices

Before leaving the subject of classification algorithms, we should briefly discuss how the performance of a particular classification or supervised learning model is quantified. For time series, the distance between observed and modelled series can be measured in some way (see Section 2.6 on various distance quantification methods). Errors in categorical classification are often summarised in the form of a confusion matrix. A confusion matrix $A$ will have elements $a_{i,j}$ equal to the number of instances with classifier $j$ assigned classifier $i$. Note that if, for $a_{i,j}$:

$$i = j \text{ number of instances correctly labelled } i$$
i \neq j \text{ number of instances with classifier } i \text{ incorrectly labelled } j

The confusion matrix is more useful than a simple misclassification count because it provides the important additional information about the distinction between type I and type II errors, and a class-by-class breakdown. For the case with two possible classifiers, positive and negative, a type II error can be thought of as a false positive and a type I as a false negative (as illustrated below). The distinction between type I and II errors is important. For example, when predicting the occurrence of dangerous algal blooms from antecedent conditions, a false positive (type II) will only result in a warning that proves unnecessary, while a false negative (type I) will not warn us of a potentially dangerous situation. Positive values in this case would also be comparatively rare.

\[
\begin{bmatrix}
\text{True Positive} & \text{False Negative (Type I Error)} \\
\text{False Positive (Type II Error)} & \text{True Negative}
\end{bmatrix}
\]

Note that values predicted to be 'True' are counted in the left hand column, and predictions of the 'False' value in the right. Actual 'True' values are counted in the top row, and 'False' values on the bottom row. The confusion matrix for the crop choice classifier introduced in Section 2.4.2 is given below.

\[
\begin{bmatrix}
8 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 19 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 2 & 153 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 & 0 & 0 & 5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 1 & 0 \\
1 & 0 & 2 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 183 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13
\end{bmatrix}
\]

\textit{Crop Choice Confusion}

Clearly, this matrix is near-diagonal, implying that performance is good for most classes. However, class 3 and 12 (flowers and root vegetables, both with 4 instances) are always
allocated to larger classes. Also note from the confusion matrix that class 8 (mango) is often incorrectly classified as class 11 (rice) and class 5 (longan). Classes 2, 9, and 13 (bean, ornamental, and tea and coffee) are completely distinct and well-labelled despite being small in size.

Most commercial data mining packages allow the use of cost matrices in the building, pruning, and evaluation of classification models. A cost matrix is simply an array of weights allocated to each of the errors listed in the confusion matrix. They can be used to tailor the model-building process to the problem at hand, or merely evaluate a cost function on a trained model. For instance, if we wished to improve the performance of the wet season crop model for the poorly-treated small classes 3 and 12, we could increase the weights on incorrectly classified instances belonging to these groups.

[118] lists common evaluation statistics for classification models. Some of these are:

**Average error** The absolute value of the residual between observed and modelled classifier values.

**Correlation** The statistical correlation calculated between known and experimental classifier outputs.

**Relative error** The ratio of the square root of the residuals between observed and modelled classifier values to the variance of the known classifiers.

**Recall** The fraction or percentage of correctly identified values from the total number of occurrences of these values.

It is in general true that the errors incurred on the training set will be less than those that will be incurred when applying the model to an independent (test) data set. Ideally this margin between training and testing error would be small, so that one can estimate the errors likely to be generated when applying the model to data without known classifiers. To be able to classify new data with confidence is, after all, the ultimate aim of building a classification model. Unfortunately, the gap between training and testing error can be considerable, especially where models are over-fitted to the training data. For this reason, evaluation of performance by multiple methods and the use of cross-validation is recommended.

The practice of cross-validation aims to reduce the margin between performance in training and performance in prediction, and is widespread among the data mining as well as general modelling communities. To cross-validate, we begin by dividing our training
dataset into $n$ distinct partitions or folds. We will construct $n$ models, each with one fold held out of the training set. Each of these models will be tested on the held-out partition. The average of the errors (however quantified) thus generated provides a much better estimate of likely model performance in the future.

Cross-validation may be stratified or unstratified. For example in 3-fold stratified cross-validation, each of the 3 partitions are simply defined as the first, second, and third thirds of the dataset, and in unstratified cross-validation each fold contains one third of the total number of instances, chosen arbitrarily (usually without replacement). When a system varies with time, a stratified cross-validation analysis will reveal changes in the model for successive partitions, but is in this case a part of the analysis itself, not an ideal indicator of model performance. Some researchers take the process one step further, eg [57], and use multiple cross-validations with different folds but for most purposes a single repetition of the cross-validation process suffices.

Another tool for evaluating binary classifier performance is the Receiver Operating Characteristic (ROC) curve ([116]). A generalisation for multi-class problems was reported in [59], but for the sake of simplicity we shall consider only the two class case here. The ROC approach has the advantage of independence from classification costs. Where the classifier output takes the form of (or can be modified to take the form of) a probability of the instance belonging to class A, the threshold $t$ above which an instance is assigned to that class is varied as a parameter. For each $t$, a confusion matrix is generated and true positive rate plotted against false positive rate. The result is a ROC curve. Ideally, ROC curves should lie as close as possible to the top and left boundaries of the unit square. For example, the classifier generating the red curve in Figure 2.5 performs better than the one corresponding to the blue. If the ROC curve of a classifier is above all other generated ROC curves, then that classifier performs better than the others on the given problem. Often, there is no such clear victor and the best classifier is chosen by considering a restricted range of $t$, or taking the Area Under Curve or AUC as a statistic for comparison.

### 2.5 Rule Extraction

#### 2.5.1 Rules

Rule extraction is often an example of unsupervised learning, although many algorithms can also be applied in a supervised manner, as a classification method. In the unsupervised
cases, a target is not explicitly defined and the rule discovery algorithm searches for any frequent, interesting, or unexpected trend in the database. The extracted rules need not cover every instance in the training set.

The basic rule takes the form \( A \Rightarrow B \), where \( A \) and \( B \) are just events of a certain type, and means that if \( A \) occurs then \( B \) occurs within some interval \( T \). Note that \( T \) could be considered a geographical space or a time interval if appropriate. \( A \) and \( B \) do not necessarily have to be the same variable— for example \( A \) could be a rainfall event type and \( B \) a flow event. Extensions of the basic rule format are possible with little more computational effort. For example, the rule \( A^1, A^2, \ldots, A^H \Rightarrow B^{T_1, T_2} \), could be interpreted as: if \( A^1, A^2, \ldots, A^H \) all occur within time \( T_1 \), then \( B \) will occur within time \( T_2 \) ([31]).

For the rainfall intensity example discussed in Section 2.4.1, rules could also be used to identify the characteristics of intense rainfall events in terms of daily climate variables. Some rules describing intense events for the city of Brisbane are:

\[
\begin{align*}
\text{Rain} < 10 \text{mm/day} & \Rightarrow \text{Not intense} \\
\text{Rain} > 42 \text{mm/day} & \Rightarrow \text{Intense} \\
\text{Rain} > 25 \text{mm/day AND RelHum} > 86\% & \Rightarrow \text{Intense} \\
\text{Rain} > 19 \text{mm/day AND MinT} > 14.3\C & \Rightarrow \text{Intense} \\
\text{RelHum} > 85\% \text{ AND StrWind} = 0 \text{ AND 34 < Day} \leq 66 \text{ AND Haze} = 0 & \Rightarrow \text{Intense} \\
\text{Thunder} = 1 \text{ AND StrWind} = 0 \text{ AND GrassT} > 18.6\C \text{ AND Day} < 63 & \Rightarrow \text{Intense}
\end{align*}
\]
Where classification in general attempts to build a model that gives the correct classifier for every instance of the training data, the rules may represent general trends within the dataset that may not be true in 100% of cases. Note that the list of rainfall intensity rules given above do not necessarily cover all possible inputs from the training set. What we get from these rules is a partial model composed of common co-occurences. For a given event type C, we may not necessarily have any rule involving C on the left- or right-hand side of the expression, but a partial model summarising the strongest trends in the database.

A partial model may have some application in missing data replacement as well as systems analysis. For example, if we discover from our data that A ⇒ B with some acceptable confidence, say 70%, (if A, then B occurs in 70% of cases), any missing segments immediately following an occurrence of A can be replaced by B. In this way we may be able to patch some of the holes in our dataset, in a way that preserves the prevailing structures in the series.

### 2.5.2 Apriori- an Example Algorithm

Developed for consumer transaction mining by Agrawal and Srikant ([3]), the Apriori algorithm is an excellent association rule extractor for use with very large categorical datasets. Association rules were developed to represent variables in a dataset that frequently occur together. Consider the database of all transactions made at a particular supermarket. Each customer (record or instance) buys a small number of items (attributes) from a very large number of possible products. The aim here is to discover what typically is and is not bought with what. The purchased items are called a 'market basket'. We are interested only in patterns that occur frequently, so the key to the problem is to find the frequently occurring market baskets and frequent itemsets within them. [26] describes the Apriori algorithm with the following:

1. Determine the support (see below) for all single attributes (itemsets of cardinality 1, ie having a single element) in the data set.

2. Delete all the single attributes that are not adequately supported.

3. For all supported single attributes construct pairs of attributes (itemsets of cardinality 2).

4. If no pairs exist end, otherwise determine the support for the constructed pairs.
For all supported pairs of attributes construct candidate itemsets (see below) of cardinality 3 (triples).

If no triples exist, end, otherwise determine the support for the constructed triples.

Continue until no more candidate sets can be produced.

To understand this algorithm there are two important concepts that we need. The first, a candidate itemset, is simply a possible market basket, a group of items that occur together. The second, support, is a measure of the frequency of a particular candidate set. For the association rule \( A \Rightarrow B \), meaning that if item \( A \) is in a market basket, \( B \) will also be in that basket, the support is simply the number of times \( A \) and \( B \) occur in a record together. Order may be preserved or not, as dictated by the problem. In supermarket basket analysis, it does not matter which of \( A \) or \( B \) is listed first, and support \( A, B = \text{support } B, A \), but when considering multiple time series, for example, \( A, B \) distinctly different from \( B, A \) and the supports for the two candidates may differ.

Once the frequent itemsets are known, the building of rules is simple, especially when we take into account the fundamental principle on which the Apriori algorithm is built: if a subset of a candidate itemset is not frequent, then the candidate itemset will not be frequent. We must of course define a threshold support to define 'frequent' and 'non-frequent', with consideration of the quantity and strength of rules we want.

Another important concept in rule discovery is confidence. For a rule \( A \Rightarrow B \), confidence is usually defined as the count of \( A, B \) divided by the count of \( A \). A high confidence implies that if \( A \) occurs, we can be fairly sure that \( B \) will also occur. The case where \( A, B \) and \( A, C \) are both frequent may come up, but both \( A \Rightarrow B \) and \( A \Rightarrow C \) will have confidence less than 50%. Note that confidence of \( A \Rightarrow B \) does not necessarily equal \( B \Rightarrow A \).

Fortunately, the Apriori algorithm is easily parallelisable. This is an important factor when we consider that the advent of barcodes and barcode scanners has made transaction databases of supermarket chains among the largest databases in the world.

### 2.5.3 A Second Rule Extraction Algorithm- ITRULE

Although ITRULE ([138]) can be used for classification, it also captures relations within variables other than the classifier. ITRULE is a bottom-up method, starting from very general rules and progressively specialising each rule until it is useful, and generalising if the accuracy of the rule becomes too low. Rules may be dropped if they do not evolve into something interesting or useful.
Both ITRULE and Apriori association rule extraction methods take only categorical data as input, as do most algorithms in this family. There are rule extraction algorithms that require a categorical target variable to be specified (note that these can be thought of as supervised learners), but deal naturally with real valued attributes otherwise. The CN2 algorithm is applied to this situation in [39]. Unlike Apriori, ITRULE makes an explicit attempt to find useful rules with each and every attribute on the right-hand side. It also performs more of an exhaustive search through the space of possible rules, and the evaluation of possible rules takes complexity into account, as detailed in [44].

2.5.4 Rule Selection

With such large databases, the number of rules generated by Apriori or another association rule extraction method may be large, often in the thousands. This is too many to be processed by a human, so we must find ways of deleting irrelevant, obvious, or otherwise unhelpful rules. Most algorithms will only output rules with confidence and support over a given threshold, but these may need to be pruned further before they are useful. Some other factors to consider are rule complexity, unexpectedness, the probabilistic relation between the elements in the rule, and importance of attributes involved. An excellent review of interestingness measures can be found in [48], and the discussion on tree pruning in Section 2.4.4 is also relevant.

2.6 Dominant Mode Analysis, Discretisation, and Time Series Similarity Measures

The concept of similarity (or inversely, distance) is important in clustering algorithms and also for evaluating model fit to known data. Because a cluster should consist of a set of points that are similar to each other and dissimilar to those outside the cluster, we must find some quantitative measure of likeness or similarity. The Euclidean distance or $L_p$ norm ($p^{th}$ root of the sum of all discrepancies to the power of $p$—Euclidean distance is the $L_2$ norm) can be used, but this will not detect similarity where series are the same rough shape but slightly out of phase, different in magnitude or start from different base levels. Depending on the problem at hand, this may or may not be desirable. As discussed in Section 2.3, offsets and magnitude differences can be removed by normalising the series, but again we must consider the intended application. Quantitative distance measures can
be used in clustering and case-based reasoning, as well as performance evaluation.

In this section dominant mode analysis and series summarisation are also considered. Not only are these techniques often useful as analytical tools, but dimensionally reduced representations can also be used as bases for distance measures. A great deal of work has been done in the area, but as noted in [75], time series data mining still faces significant challenges. Example discussions of time series data mining with respect to distance measures and clustering can be found in [72], [79], [10], [81], and [76]. Times series analysis with support vector machines (see Section 2.4.8) is detailed in [126], and the problem of extracting distinctive patterns in multivariate time series is addressed in [111].

### 2.6.1 Discretisation

Where data is numeric, clustering can be used to translate real values to categorical by replacing raw instances with a label indicating the covering cluster. The resulting compacted representation can then be passed to further processing or rule extraction. Many other discretisation methods also exist, ranging from the very simple histogram binning to the complex.

Translation from continuous numerical to discrete data reduces the complexity of the variable in question and renders it more amenable to analysis with most classification algorithms and many visualisation techniques. Some loss of information will occur, but where data is noisy, it may be appropriate to decrease resolution by discretisation. In this way the level of real information content is more accurately reflected. For time series, smoothing methods may be used for the same purpose without discretising, but the consideration of, for example, conditional probabilities is far simpler where data is discrete or categorical. As with any pre-processing techniques, advantages and disadvantages must both be considered, and some experimentation may be informative. Choice of an appropriate resolution in the discretised variable must also be made with attention to the problem at hand and quality of input data.

### 2.6.2 Text Distance

Although not directly relevant to environmental data, which is largely numeric or categorical (e.g., presence/absence), the problem of text distance measures warrants discussion if only to illustrate the way in which difficulties are dealt with. Of course, categorical representations use an alphabet of their own, so the concepts developed for text matching
can be applied to categorical problems where appropriate. The applications of measuring similarity between text strings are widespread, web searches being one active area of application. The text string similarity problem provides a good illustration of some of the issues involved in distance calculation, especially when we consider that letters may represent other entities, such as the presence of an algal bloom or the stages in release of water for irrigation.

We will begin discussion of text similarity by noting that the letters of the alphabet are not ordered, ie the distance between the letter a and the letter z is the same as the distance between a and c. Thus we may simply express the distance between two sequences of length n as

\[ D(\text{string}1, \text{string}2) = \frac{1}{\sum_{i=1}^{n} d_i} \]

where \( d_i = 0 \) if the \( i^{th} \) character matches and \( d_i = 1 \) if not. Differing length of sequences also needs to be accounted for. We can, of course, simply take the length of the longer sequence as n and count the blank spaces as another character, but this may not be adequate.

Consider the two strings spaces and paces. Intuitively, these two sequences are very close, but the method described above returns a distance of 1, or no similarity at all. A better measure is the replacement, or the number of characters that must be altered, added, or deleted to transform one sequence into the other, normalised by the larger number of elements. Note that \( D(\text{string}1, \text{string}2) = D(\text{string}2, \text{string}1) \) as we would hope. Replacement distance \( D_r(\text{spaces}, \text{paces}) = \frac{1}{6} \). Another text distance measure that rates our example strings as close is the Longest Common Sequence or LCS measure. \( D_{LCS}(\text{paces}, \text{spaces}) = \frac{5}{6} \). Note that this is more properly a similarity measure (\( D_{LCS} \) increases with similarity, rather than \( D_r \) which increases with dissimilarity or distance), and that \( D_{LCS} + D_r \) is not necessarily unity because of the possibility of deletion without replacement in the \( D_R \) measure. The case of \( D_r(\text{mists}, \text{miss}) + D_{LCS}(\text{mists}, \text{miss}) = \frac{1}{5} + \frac{3}{5} \) is an example of this last point. This may seem to imply that \( D_r \) is a better distance measure, but the value of common sequences should not be ignored. The fields of human language analysis, where multiple characters may represent one sound, or DNA analysis (a field where text-style similarity is of vital importance), serve to demonstrate this.

Other measures can be used to compare the similarity of time series. Usually, the basic approach (as with Fourier and wavelet analyses) is to map the series onto some low dimensional space and then apply simple distance measures like the Euclidean distance between the mapped vectors ([54]). See Section 2.6 for a discussion of similarity/distance measures between relatively low dimension entities. Both wavelet and Fourier spectra can
be compared in this way.

2.6.3 Data Reduction by Extraction of Dominant Modes

The concept of extracting dominant modes of a dataset is well established. Like the Principal Component Analysis (PCA) briefly discussed in the distance measures section above, the aim is to reduce dimensionality. Most techniques involve decomposition by basis functions, usually but not always orthogonal. These are often useful for dimensionality reduction in large databases with many non-independent attributes.

Other methods consider attributes individually. Fourier analysis is the best known and oldest of these (many works have been written on the topic, for example [9]). Using a hydrological example, Fourier transformation can take a time series of streamflow values to a set of frequencies with amplitudes. The process can be thought of as decomposition into a collection of additive sine waves, and so works best for series with an underlying periodic (e.g., seasonal) signal. Fourier analysis deals naturally with data where components (for example flow peaks) are not well separated. In these circumstances, analysis of individual peaks can be difficult, because of the absence of an extractable tail of the flow response to a single rainfall pulse. However, Fourier transforms retain no information about positioning of events in the series.

The full Fourier spectrum of a time series of length \( N \) contains \( N \) elements, although many of these may be zero or near-zero in magnitude. It is not therefore useful to compare entire spectra. Instead, the vector of coefficients is truncated to contain only the \( n \) elements of greatest magnitude. Returning to the South-East Queensland flow database introduced in Section 2.3, Figure 2.6 shows a 1096 point daily flow record, with inverse Fourier reconstructions of varying \( n \). These were computed with the standard Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) algorithms. Clearly, most of the information contained in the hundred coefficients greatest in magnitude is in the first few, and even using 601 out of 1096 candidates does not produce a good approximation to the series.

The periodic signal in this series is weak. Although major flow events seem to occur yearly, the magnitude and nature of the high-flow season is highly variable. Figure 2.7 is the result of a similar analysis on a series of 837 average weekly maximum temperatures from the Adelaide West meteorological station in South Australia. Note that \( n \) ranges from 1 to 26 rather than 1 to 601. In this (highly periodic) case, Fourier spectra do provide an avenue for good series compression.
Figure 2.6: Example Fourier Reconstruction of a Flow Series with Low Periodic Signal

Figure 2.7: Example Fourier Reconstruction of a Temperature Series with High Periodic Signal
2.7. PREVIOUS APPLICATIONS

More recently a different, although related, method has emerged. Wavelet transformations spatially decompose the signal into a series of wavelets, the characteristics of which can be carefully tailored. Temporal information is retained, and there is the additional advantage of detailed analysis at varying time scales ([9]). Wavelet transformation has been used to group catchments by taking the wavelet spectra (a form of the results of the wavelet transform) of streamflow records as a signature of catchment response characteristics and clustering on these ([163]).

There are also various series comparison techniques, some of which are probably inappropriate to hydrological problems. Piecewise linear representation is unlikely to be helpful due to the spiky nature of long-term hydrographs. Dynamic Time Warping, which allows stretching or compression of time axes to achieve better fit between sequences, may not preserve information valuable to hydrological analysis, but has found application elsewhere. Recently, methods for image database searching have been developed with various mathematically rigorous shape matching bases, which may find application in diverse areas. For an example, see [22].

2.7 Previous Applications of Data Mining to Environmental Science

In this section we will briefly consider the body of environmental problems that have been addressed with the aid of data mining techniques. While most environmental scientists are unlikely to adopt machine learning techniques over more problem-specific modelling techniques, some work has been done in the area. A few specific papers are referenced, but as with all brief literature reviews we cannot hope to mention all the available material.

The University of California at Irvine hosts the UCI Machine Learning Repository, which houses a large collection of standard databases for testing data mining algorithms. Among them are databases on the abalone age prediction from physical dimensions, ionosphere radar returns and soybean disease identification. A small number of research groups also exist specific to artificial intelligence or data mining in the environmental sciences. The D2K (Data to Knowledge) project by the Danish Technical Research Council works in the field of hydrology, and the BESAI (Binding Environmental Sciences and Artificial Intelligence) Working Group is based in the Universitat Politècnica de Catalunya in Spain, is building a suite of software for intelligent environmental data analysis (GESCONDA) ([52]). Four International Workshops on Environmental Applications have been held over
CHAPTER 2. REVIEW OF DATA MINING TECHNIQUES

the period 1997–2004, producing some of the papers discussed below. The output of the latest workshop, which focussed on genetic algorithms and neural networks, is discussed in [121].

A number of papers have been published advocating or suggesting the use of machine learning techniques in environmental sciences. Babovic’s short letter to this end in Hydrological Processes [7] specifically mentioned Genetic Programming, where mathematical models are built under evolutionary conditions. The techniques were applied to vegetation-induced flow resistance in [8]. Knowledge-based artificial intelligence systems were advocated for environmental use in [60].

Data mining and machine learning are of course not restricted to the methods discussed here, and some less common techniques have been applied to environmental problems. In [124], Hidden Markov models were used to model rainfall patterns over Brazil with interesting results, and [99] applied qualitative reasoning to meteorological problems. Cloud screening for meteorological purposes was also investigated with Markov Random Fields in [17]. Sudden death of oak trees was modelled with support vector machines in [55]. Genetic programming was used to model glider possum distributions [157], and [34] used genetic algorithms for attribute selection in benthic macroinvertebrate modelling. Decision trees were then built from the reduced dataset.

The k-means algorithm discussed in Section 2.3 finds application in [163], where 286 Australian streamflow series were clustered according to wavelet spectra (see Section 2.6). The aim was to identify groups of catchments with similar physical characteristics. Clustering or similar methods have also been used for hydrological regionalisation in [45] and [127] for study areas in the United Kingdom and Northwestern United States respectively. Clustering was also applied to cyclone paths in [18]. The mammalian genome was clustered using a fast, novel method ([25]). The hierarchical clusters generated therein were then compared to two competing evolutionary hypotheses, and found to provide evidence for the Marsupionta hypothesis. Clustering was also used in [151] to cluster water samples.

The United States Digital Government Research Center funded a project with the direction of integrating spatial data mining approaches with other modelling and data-manipulation techniques to create a solute transport model, the proposal for which can be found at [21]. Data mining techniques have also found application in spatial problems such as modelling fish distribution ([147]) and soil erosion patterns ([42]). Soil erosion was also tackled with classification in [133], as were other soil properties in [90]. In [133] a comparison of results obtained from a classical statistical approach and classification
2.8. CONCLUSIONS

algorithms is also made. Regression trees and generalised linear models were both used in [90] with a view to obtaining system information. This paper is also a good example of rigorous data treatment- the authors discuss sampling techniques and data collection and treatment in some depth. Case based inference was used for spatial modelling of species distributions and habitats in [122].

Various classification algorithms have been applied to a wide variety of non-spatial environmental problems as well. The problem of extracting rainfall intensity information from daily climate data ([142], [146], discussed in Section 2.4.1) was initially addressed using a number of classification methods. The C4.5 decision tree building algorithm (see Section 2.4.2) was used in [40] (also mentioned in Section 2.4.2), where choice of crop was modelled as a classification problem with considerable success. This example was discussed in the body of this paper. Mosquito population sites were classified in [148], with a view to controlling the spread of malaria. Agriculturally, classification has been applied to apple bruising ([64]), mushroom grading ([30]), bull castration and venison carcase analysis in [162], and perhaps most famously in Michalski and Chilausky's classic soybean disease diagnosis work [92]. Regression trees were applied to sea cucumber habitat preference modelling in [38].

The use of rule learning for the environmental sciences is discussed in [123]. The example in this paper is the state identification of a wastewater treatment plant. Rule learning was also used to investigate a streamflow/electrical conductivity system in [144]. In [39], the process of rule learning is illustrated with examples from water quality databases and the CN2 algorithm.

Artificial Neural Networks have been adopted widely by the environmental science community (for example see [78] and [5] on nitrogen loading, [89] on deforestation, or [51] and [50] on water quality, or the discussion in [121]), and it is for that reason that we do not describe them in the paper. Classification and clustering are beginning to be used in some sectors. We hope that rule-based learning and other data mining methods will follow suit as more researchers become aware of the potential that exists in the large reservoir of techniques that have been created in recent years.

2.8 Conclusions

In this chapter we have discussed a number of data mining concepts and techniques. Where possible, factors such as complexity, restrictions on data types, and model interpretability
have been noted, as they are important concepts to consider when choosing a data mining algorithm. Performance quantification method such as the confusion matrix were also introduced. Throughout subsequent chapters, reference will be made to the algorithms and methods discussed here.

The chapter was also intended as an introduction to the spirit of data mining as a science, and to demonstrate the breadth and variety of algorithms available for data analysis. To this end we have described several algorithms that we will not use in later chapters, but which readers may wish to investigate further if they seem appropriate for their desired uses.
Chapter 3

Identifying Vulnerability to Gully Erosion in the Ben Chifley Dam Catchment

3.1 Statement of the Problem, Motivation, and Data

In the Ben Chifley Dam Catchment (BCDC) in New South Wales, as in large parts of Australia, gully erosion is a significant land quality issue. Most often appearing on grazing land, the formation of gullies can result in soil losses and decreased quality of receiving waters. We would like to better understand where and under what conditions gullies are most likely to form, so management for prevention and treatment can be targeted to these areas. In the hope of achieving a better understanding of gully erosion processes, we will examine data from the BCDC. Gully erosion impacts the water quality in the streams and dam, which supply water to the city of Bathurst.

A dataset of about 2000 instances was generated and analysed in a previous study ([136]), where regression analysis was used. Each instance consists of a spatial cell with uniform soil type, underlying geology, land use, slope, elevation. Each one corresponds to a homogeneous irregular polygon on a GIS overlay map (see Section 4 of [136]). For each polygon, area covered and length of gully present within the boundaries are also listed. In [136], length of gully in each cell was normalised by the total polygon area and called gully density ($GD$), expressed in metres per square kilometre.

Our aim is some model or expression for likelihood of gully erosion in terms of the descriptive features listed above. First, we shall investigate the suitability of $GD$ as a
measure of gully occurrence by visual means. As shown in Section 3.2, it is necessary to modify the existing measure to obtain good results. Reasons for this are discussed in that section. We also discretise the gully density measure, so that certain methods can be applied.

With a new representation of gully density, we shall then examine the relationship between this variable and the descriptive variables. Because the dataset is small, and the number of variables low, it is practical to do this graphically. It is found that significant information about gully density can be obtained by consideration of single variables, in particular soil class and elevation.

Four data mining techniques are then applied— the very simple OneR algorithm (see section 2.4.7), and the C4.5 and ADtree decision tree extraction algorithms (described in Sections 2.4.2 and 2.4.8 respectively). OneR splits a single variable in sections where the corresponding target (gully density) values are homogeneous or near-homogeneous. It provides a measure of the information about the target that can be extracted from a single variable, and serves as a point of comparison (benchmark) for the more complex methods.

Decision trees are well equipped for graphical presentation of results, which is the reason they were chosen for this project. These models are easily understandable and interpretable by non-experts. For an example tree, see Figure 2.3 in the previous chapter, and the accompanying discussion. OneR results and trees for gully erosion are extracted and given in Section 3.4, which is followed by a general discussion of the results obtained.

### 3.1.1 Data

A brief description is given below (after [134]) for all variables.

**Area** is a real variable in square kilometres. Range 0.5–14.6.

**Landuse** takes one of four land usage categories- 1 Cropping, 2 Native and Plantation Forestry, 3 Pasture, and 4 Horticulture. Overwhelming majority of class 3 (Pasture).

**Geology** contains ten categories. Unevenly distributed between the following groups:

1. Carboniferous granite intrusions
2. Devonian lithic sandstone from felsic volcanics interbedded with siltstone and fine sandstone
3. Ordovician shale and turbidite sequence
3.1. STATEMENT OF THE PROBLEM, MOTIVATION, AND DATA

4 Ordovician intermediate to mafic volcanics
5 Ordovician mafic intrusions
6 Ordovician and Silurian mafic intrusions
7 Permian sediments
8 Quaternary alluvium
9 Silurian sediments
10 Silurian grano-diorite and granitic intrusions
11 Silurian mafic intrusions
12 Tertiary basalt, gravel etc

Soil type is unevenly distributed between the 11 categories given below.

1 Alluvial soils
2 Kranozems
3 Chocolate soils
4 Non-calcic brown soils
5 Red earths
6 Red podzolic soils
7 Shallow soils
8 Skeletal soils
9 Siliceous sands
10 Yellow earths
11 Yellow podzolic soils
Slope is discretised into ordered categories as illustrated below by a list of upper bounds. Values for this variable and for elevation were not directly measured, but were derived from a Digital Elevation Model (DEM). All non-discretised values are given in degrees.

1 0
2 2
3 4
4 6
5 10
6 70

Elevation ranges from 690 to 1340 metres above sea level in the BCDC. The data has been discretised to a rating between one and ten, with upper limits as listed below in metres above sea level. Both Slope and Elevation were discretised manually, in a manner such that each category contained a similar number of cases while maintaining a physically meaningful range.

1 700
2 800
3 850
4 900
5 950
6 1000
7 1050
8 1100
9 1200
10 1400
3.2. **APPROACH- GULLY DENSITY**

Gully length is the total length of all gullies in the cell. In metres, it ranges from 0 to 10072.

There are a number of issues that must be addressed as we consider this data. The first is the highly uncertain or qualitative nature of the data we have. While Slope, Elevation, Landuse etc can be determined well, they are point measurements in a continuous medium-we must rely on the GIS map of polygon boundaries, knowing that attributes like geological class are sparsely sampled. Also, one soil type may have properties close to the boundary of its class, and the final allocation of soil type class was likely to have been subjective. We do not have exact soil composition data (of course, we do not necessarily want this data-it is complex and difficult to understand for the non-expert).

Categorical attributes like soil type, land use, and geology must be represented as unordered non-numeric data, which makes the application of regression techniques problematic. In addition, Slope and Elevation have been discretised in advance such that the distribution of points between all classes was roughly even, but these two are still ordered.

The distribution of values is highly uneven across the target gully density variable and several of the descriptive features is highly uneven. For example, three quarters of the 2000 instances have no gullies at all. Thus, even a simple presence-absence analysis will be weighted towards the null $GD = 0$ case if we do not take steps to prevent this-as discussed in Section 2.4.2 C4.5 and many other methods tend to predict the majority class. Most of the instances also have the pasture land use class.

Gully density as defined in [136] is not an ideal classifier. As cells are not defined by a grid, but as regions of homogeneous characteristics, cell area varies significantly. Cells with a very small area and a small gully formation or part of a gully intersecting the boundaries have an exaggerated density, as acknowledged by the authors of that study. We shall see this illustrated graphically in the next section. The existing gully density values also vary through several orders of magnitude, which is not desirable as it may lead to bias. Accordingly, we will modify the measure.

### 3.2 Approach- Gully Density

The first task at hand is to properly define a target variable. There are three routes we could take to improve the existing gully density measure. Most simply, we could introduce a threshold area below which a cell is disqualified from analysis or treated differently. Or, we could re-define the entire function taking area and total gully length to density. This
could have the advantage of forcing a narrower spread of values of the target variable. A scatter plot of gully density \( GD \) against cell area is given in Figure 3.1. The bias towards high density with small area is clear, manifesting as asymptotic-like behaviour on the vertical axis. As we can see, this artefact accounts for all the densities above a certain level.

By excluding from analysis all instances with \( \text{Area} \) less than 0.5 km\(^2\), the behaviour almost vanishes. We then obtain the data illustrated in Figure 3.2.

While there is still a substantial increase in density with area close to 0.5, the number of points also increases approaching the threshold, so we will allow this on the grounds of distribution. That is, the more points there are in a certain region, the more likely it is that some high values will occur. Also, the number of instances is now 399 where the raw
3.3 VISUAL INVESTIGATION

dataset contained almost 2100, so we are naturally reluctant to reduce the set further.

Now, GD can be considered as a continuous variable, or discretised in some way. Both
these options will be used for different algorithms. Three discrete classifiers will be defined
with categories as follows:

1 Absent/Present
2 Absent/Low/High
3 Absent/Low/Med/High

The Absent category is all zero densities, with all the other classes defined so as to
contain an equal number of instances. This equi-cistribution mechanism is consistent with
our principle of avoiding bias where possible, and is also an intuitive choice.

3.3 Visual Investigation

As stated in Chapter 1, visualisation is an important preprocessing step. Had we (and [136])
bypassed a visual examination of the target variable and proceeded straight to application
or a data mining or regression algorithm, it would be unlikely that the problem with GD at
small values of Area would have been picked up and dealt with. Poor or misleading results
may then have been obtained. Visualisation should be an initial step for all analysis
projects. Here, with seven variables, it is possible and practical to visually examine all
variables, but where this is not possible, an examination of a subset of variables may be
appropriate. See Chapters 7 and 8 for a discussion on selecting a representative group of
variables.

A few of the scatter plots we can generate with gully density measures and a single
descriptive feature show a clear or useful trend or structure. Figure 3.3 shows soil class
against gully Absence/Presence, and we can see that a Soil class of 6 or 9 (red podzolic
soils or siliceous sands) implies a likelihood of gully presence, where members of classes
2, 5, and 10 (krasnozems, red earths, and yellow earths) contain very few gullies. This
information was completely obscured before the removal of cells with small area.

Other trends that we can identify visually are given below:

Elevation Chance of gully absence increases with rising Elevation class. Any Elevation
category above 6 (greater than 1000 m above sea level) is unlikely to have gullies present
at all, and the very low classes (1, 2, 3) (less than 850 m above sea level) will almost
certainly imply the presence of gullies. These low classes also have a large proportion of high gully density.

**Geology** Classes 3, 5, and 11 (ordovician shale and turbidite sequence, ordovician mafic intrusions, and teriary basalts) are unlikely to have gullies present, but in categories 2, 4, and 8 (devonian lithic sandstone, ordovician intermediate to mafic volcanics, and quaternary alluvium) about half the cells show gully presence. This is also true of class 9 (silurian sediments), with the additional information that most of the gully densities in these cells are severe.

**Soil** As noted above, classes 2, 5, and 10 have few gullies, and more than half the cells in class 6 and class 9 are high in gully density. In addition, 8 and 11 (skeletal soils and yellow podzolic soils) also have a strong gully presence.

**Land Use** Most of the cells with a land use of category 2 (forestry) have no gullies.

**Slope** The density levels are approximately evenly distributed over all slopes. Therefore, it is unlikely that useful information can be extracted from this slope data unless it becomes viable when combined with the information contained in another feature.

This may seem like a comprehensive list of the behaviours of gully density with different variables, but most of the well-defined gully density levels are determined by the less populous classes of explicatory variables- most instances do not fall into these classes. For example, in Figure 3.3, we can see that many instances do not belong to one of the soil types 2, 5, 6, 9, and 10 where gully density is well-defined (ie where Figure 3.3 tells us with a high degree of certainty whether gullies are likely to be either present or absent). Also, where we have stated that a large proportion of a given class belongs to a particular non-
zero density level, we typically mean a value of 40-50%, which does not uniquely determine the class value. The rules concerning elevation class are the exception to this observation, and are shown in Figure 3.4. For example, note that the top two elevation classes contain almost no cells with any gullies present at all.

The other two-dimensional visual tool we may use are the plots generated between pairs of descriptive features. However, while a detailed investigation of the matrix of these plots may reveal much about the nature of BCDC (for example, revealing relationships between soil class and geology) it does not tell us much with bearing on our gully density problem. The real use for this visualisation technique will be in the post-processing of the rules generated by the data mining algorithms applied in the next section, when we will need to examine the physicality of generated rules and models to see what can be simplified and what may be an artifact of the method.

3.4 Application of Data Mining Algorithms

The aim of this case study is to produce results that can be easily understood by a catchment manager or farmer, so the obvious place to start is with decision tree and alternating decision tree methods, and rule extractors. These methods produce easily understandable models (see Chapter 2). We will compare results from two different decision tree builders, the alternating decision tree algorithm ADtree [49] and the much-used classic decision tree extraction method C4.5 [117]. Both of these are well verified techniques, and C4.5 in particular has been widely applied in the data mining field and has enjoyed considerable
research attention (see for example [120], [34], [41]). C4.5 was selected from the available pool of decision tree discovery algorithms because it is considered as an industry standard, and ADtree was chosen because it produces additional information such as confidences at each leaf.

In addition, a modified version of C4.5 that produces a rule list (j48 PART) was applied, to obtain a slightly different model and hence a new perspective. The OneR algorithm (see Section 2.4.7) will also be applied. Recall from Chapter 2 that this provides us with a benchmark, and quantifies how much knowledge can be gleaned from the most informative class. We know from the visual investigation the best variable is probably Elevation.

Beginning with the simple gully Absence/Presence classifier, with instances edited as discussed in Section 3.2, we first use the OneR algorithm on the elevation feature with the usual 10-fold cross-validation procedure as described in Section 2.4.9. We find the simple classification model

\[
\text{Elevation}\text{Class} \leq 6 \rightarrow \text{Absence} \\
> 6 \rightarrow \text{Presence}
\]

As listed in Table 3.1, this rule correctly predicts 80.2% of instances. Using the C4.5 algorithm as provided in Weka (see [1:2]), and optimising the appropriate inputs, the best tree structure we can find has 86.7% predictive accuracy and 19 nodes in total (see Table 3.1). To obtain this tree, we enforce binary splitting on categorical, non-ordered features (Geology, Landuse, Soil type), and use Reduced Error Pruning (REP). See Section 4.5.3 for technical details of these options. A tree with slightly higher predictive capacity (87.0%) results from binary splitting without REP, but the number of nodes is more than three times greater. The parameter controlling minimum number of instances to be covered by a leaf was ultimately left at 2, although it was varied in experiment. Most leaves contain far more instances than this threshold.

The same combination of REP and binary splitting gave us the best set of j48 PART rules. With 9 rules, predictive accuracy was estimated at 87.5%. Again, binary splitting alone resulted in an only very slightly better and much larger rule set. The nature of these rules are discussed in the next section. The ADtree algorithm, only applicable to the binary Absence/Presence classifier, resulted in a 31 node tree with 84.2% predictive accuracy.

For the other classifier schemes as well as for the above, the number of predictor nodes, and predictive accuracies estimated with 10-fold cross-validation are summarised in the
### 3.4. APPLICATION OF DATA MINING ALGORITHMS

<table>
<thead>
<tr>
<th>Number of Classes</th>
<th>Algorithm</th>
<th>% Accuracy</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>OneR</td>
<td>80.2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>C4.5</td>
<td>86.7</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>j48 PART</td>
<td>87.5</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>ADtree</td>
<td>84.2</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>OneR</td>
<td>72.0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>C4.5</td>
<td>79.7</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>j48 PART</td>
<td>79.9</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>OneR</td>
<td>62.4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>C4.5</td>
<td>73.4</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>j48 PART</td>
<td>74.9</td>
<td>44</td>
</tr>
</tbody>
</table>

Table 3.1: Best Accuracies and Number of Predictor Nodes

Tables 3.1. The breakdown of correct and false predictions shall be discussed briefly when we consider confusion matrices.

Recall from Section 2.4.9 that confusion matrices present the number of instances allocated correctly and incorrectly to each class. The better confusion matrices (listed below for C4.5) are those closest to a pure diagonal form. The matrix from the *Absence/Presence* two class problem show a near-equal proportion of incorrectly classified instances between the classes, with a slight bias towards predicting the majority (*Absence*) class. However, when the three class (*Zero, Low, High*) problem is introduced, we find a distinct confusion between the *Low* and *Zero* classes, and also to a lesser degree between *Low* and *High* classes. The *Zero-Low* confusion persists into the four class (*Zero, Low, Medium, High*) classification problem, and a similar artefact appears between high and medium classes. A C4.5 confusion matrix from each classifier scheme is given below to illustrate this.

\[
\begin{bmatrix}
193 & 28 \\
22 & 156
\end{bmatrix}
\text{Absence, Presence}
\]

\[
\begin{bmatrix}
195 & 20 & 6 \\
20 & 52 & 17 \\
3 & 14 & 72
\end{bmatrix}
\text{Zero, Low, High}
\]

\[
\begin{bmatrix}
191 & 15 & 11 & 4 \\
14 & 35 & 9 & 1 \\
8 & 6 & 32 & 14 \\
2 & 2 & 4 & 41
\end{bmatrix}
\text{Zero, Low, Medium, High}
\]
We would expect the confusions listed above, but there is another interesting feature of the confusion matrix for the four-class problem. The confusion between the lower two classes (Zero and Low) and the higher two (Medium and High) is very small. Therefore, a natural two-class classification scheme is suggested—zero and very small densities against all greater density values, or a modification of the Zero category to Zero and very small. This will be left as a suggestion for future work.

3.5 Discussion of Results

As Table 3.1 demonstrates, the best possible accuracy decreases by about 10% and the number of required predictor nodes increases as the number of classes in the problem increases. This is an intuitive trend—the more complex the problem, the more complex the solution must be.

For the four class problem, we will examine the C4.5 decision tree with binary splitting enforced, which produced the best tree for that combination of classifier and algorithm. It is given in Figure 3.5.
Figure 3.5: C4.5 Decision Tree, Four Classes
CHAPTER 3. VULNERABILITY TO GULLY EROSION

This tree also follows our visual observations. For example, the first split separates the instances into Elevation greater than six, where we observed that the majority of cells have no zero gully density, and less than or equal to six, where GD is higher. High up in both branches of the tree we also find the Soil variable split into Soil = 6 and Soil ≠ 6. Recall from Figure 3.3 that soil category 6 (red podzolic soils) is the only class with an overwhelming majority of gully presence.

However, decision trees can encode more information than the distribution plots of Figures 3.4 and 3.3. The second split on the right-hand branch is on geological class = or ≠ 2 (class 2 is devonian lithic sandstone from felsic volcanoes interbeded with siltstone and fine sandstone), which did not seem to be one strong indicator at the visualisation stage. At that point in the analysis, we were considering only relationships between gully density and single variables, but the decision tree of Figure 3.5 tells us that for high elevations, geology is has a strong influence on gully densities. At lower elevations, the action of geological class may be different, or obscured by other factors.

As noted in Table 3.1, the tree of Figure 3.5 correctly predicts 73.4% of instances. The confusion matrix given in the previous section represents the breakdown of correct and incorrect classifications. From this, we can state that the greatest confusion lies between adjacent categories, as we would hope. The model appears to be suitable for the problem, and capturing dynamics well. However, examination of the OneR rate of accuracy for this four-class problem is 62.4%, achieved only with four nodes, where the C4.5 decision tree contains 44 nodes. The extra 40 nodes improve overall accuracy by just over 12%. It is possible that some pruning of the tree could improve interpretability and simplicity by reducing the number of nodes with small or insignificant loss of accuracy.

What we have gained from the machine learning approach to this problem is essentially a formalisation and continuation of what we discovered in Section 3.3, with inclusion of multi-dimensional information. A study on the ability of non-experts to produce C.45 (see Section 4.5) style decision trees, [154], used more sophisticated visualisation tools than we have used here, shows that in circumstances with a small number of descriptive variables it is possible to manually construct trees of accuracy comparable to C4.5 with the help of sophisticated visualisation tools. These trees may also be smaller in size than their automated counterparts, but take much longer to generate.

The case of Absence/Presence gully erosion prediction, is likely to be one of those where the manual technique is viable. The strong split on Elevation suggested in Figure 3.4 would be our first choice, and the results obtained by the OneR algorithm given in Table 3.1 imply
that this first split could correctly predict up to 80% of instances. What we see in the final
tree is a show of trends similar to the observations we made from the simple visualisation
in Section 3.3.

Although *Elevation* is by far the strongest variable in terms of predictive capacity, this
fact does not fit well with physical considerations. We would not expect that elevation itself
has a strong effect on the presence of gully erosion, and a strong dependence on elevation
limits the applicability of our models to other regions, where topological structures may
be of different character. Nor does it provide insight into the physical processes of gully
formation and growth.

In conjunction with Smith, it was concluded that *Elevation* is probably acting as
a surrogate for another variable or variables that do have a causal effect. Following the
method of [63], we set about discovering the relationships within the descriptive variables by
modelling each feature with all the others under a C4.5 scheme. No conclusive dependencies
were discovered. This is not surprising, because if the causal variable existed in the dataset,
its direct relationship with gully density would have been identified.

*Slope* is a strong candidate for the true physical properties represented by *Elevation*. From
physical considerations, one would assume that *Slope* and *Elevation* would be
strongly correlated in some way, but the visualisation and cross-modelling processes yielded
no evidence confirming this. With Smith, the conclusion was reached that the *Slope*
variable presented in this dataset is of poor quality. It is known that certain difficulties were
encountered in the calculation of this variable ([134]), and it does not hold some of the
properties one would expect from physical considerations (such as a relationship with
*Elevation*, which is a far simpler variable to estimate from a DEM).

A number of different samples of the raw slope data were tested and experimented upon,
but none shed any further light on the problems of *Elevation* as a possible surrogate, or
on the gully prediction problem as a whole.

### 3.6 Comparison with Classical Techniques

The model we see represented in the C4.5 decision tree in Figure 3.5 and the other models
generated for multiple gully density classes agree well with the results obtained by Smith in
[136]. The data in that study did not have the small area treatment (Section 3.2) we made
to filter out unrepresentatively large gully density values. Thus, the data was too noisy
to obtain informative visualisations like Figure 3.4, and complex statistical techniques
were employed. Analysis primarily focussed on the soil and geological variables. Two-dimensional decision tables were used in conjunction with other methods, details of which can be found in the publication.

Among the trends identified in [136], soil classes of 2, 3, or 5 (krasnozem, chocolate, and red earth) were also seen to contain few gullies. This is comparable to our finding that soil classes 2, 5, and 10 (krasnozems, red earths, and yellow earths) had fewest gullies, although the source of this finding, Figure 3.3, was constructed from absence/presence data rather than a numerical GD value. High gully density soil classes identified included class 6 (red podzolic) as noted from Figure 3.3.

Low gully geological categories were basalt, silurian granodiorite and granitic intrusions, ordovician mafic intrusion, and the ordovician shale and turbidite sequence (11, 9, 5, and 3). There is a Geology \(=, \neq 11 \) decision high on the left branch of the tree in Figure 3.5 with a strong GD = 0 leaf, but splits on other variables listed above are not as prominent.

Soil categories 1 (alluvial soils) and geological values 8 (quaternary alluvium) and 4 (ordovician intermediate to mafic volcanics) combined to produce a high gully density (or risk, as defined by [136]), as did red podzolic soils with geological categories 1, 8, and 5 (shale). These combinations are not immediately evident in Figure 3.5. More trends are given in [136], some of which are similar to those identified here, and others which are not. The two studies have taken differing perspectives, and obtained different information. It is not likely that one presents better results than the other. Rather, they are both valuable sources of knowledge for catchment managers and other interested parties.

### 3.7 Conclusions

This study produced three main points: the importance of visualisation as a modelling step, the necessity of filtering of input data and the predictive power of Elevation in gully density modelling for the Ben Chifley Dam Catchment. The study illustrates that on this occasion, the informative visualisations obtained only from filtered data (for example Figure 3.4), were a rich source of information. The relative simplicity of our approach was made possible by this visualisation and the preprocessing step. Some of the system knowledge obtained from this analysis reflected that extracted in [136], but the data mining approach is less labour intensive and produced easily interpretable graphical results. All variables could be examined and treated automatically by the decision trees.

The predominance of Elevation as a predictor, together with the lack of findings from
the feature dependence discovery exercise, leads us to believe that the data representing slope of a given polygon is flawed. This may be due to inadequate resolution of the generating DEM. We conclude that *Elevation* is acting as a surrogate variable for *Slope*, and slope may in fact be a driver of gully erosion in the Bathurst area. However, without good slope data, we cannot confirm this theory or state what the relationship between slope and gully occurrence is.

Both the modelling processes in this study and in [136] resulted in information about gully likelihood with *Elevation, Soil, Geology* and other variables in the Ben Chifley region. Some insight was gained into the nature of the problem and the data as given. With more accurate slope data, we are confident of producing a gully erosion severity prediction scheme with the potential to be regionalised to other areas.
Chapter 4

Classification of Rainfall Intensity

This chapter details the second example investigation of the utility of data mining techniques for environmental problem solving. We will use classification algorithms to build models returning rainfall intensity from daily climate variables. The problem is considerably more complex than the straightforward analysis carried out in Chapter 3, where the target variable was explicitly defined and the number of input variables small. There, the relatively simple structure of the problem allowed for a correspondingly simple analysis, and the dataset was provided by another study. Here, we must define a target that appropriately reflects both the available data and the goal of the analysis. The problem is well suited to classification: each instance (day) requires a rainfall intensity label. Several algorithms will be tried and compared (including the two classification algorithms used with good effect in the previous chapter), and benchmarks and parameter selection methods are used. Issues of parameter selection, data distribution, and performance evaluation also arise and are dealt with.

The investigation detailed below follows a real world problem from motivation to expression of results. Difficulties and subproblems that arise are typical of complex environmental systems modelling, and we demonstrate the concerns that should be addressed by data mining and modelling projects.

The problem is explained in Section 4.1. The relevance of intense rainfall event identification is indicated in the subsequent section. From there, we proceed in Section 4.2 to examine the data available and the data selected for modelling. We define quantitatively what we mean by an intense event in Section 4.3, and describe the algorithms used. Results of the analysis, as well as a detailed explanation of the experimental process, are given in later sections.
CHAPTER 4. RAINFALL INTENSITY

The modelling process we shall follow is as follows:

1 Definition of the problem
2 Preliminary investigation of climate data
3 Creation of quantitative target attributes for intensity
4 Preliminary application of classification algorithms
5 Detailed application of selected classification algorithms
6 Investigation and discussion of results

Note that item 3 also includes a brief comparison of the properties of the generated intensity attributes, and item 5 covers experiments such as attribute selection methods, parameter variation, and benchmarking.

4.1 Statement of the Problem and Motivation

In this chapter, we seek to identify intense rainfall events using only readily available daily climatic data. A preliminary study of this problem is presented in [142]. Most rainfall measurements are taken at daily resolution, and thus the nature of rainfall events at a finer time scale are not captured. Such capture has many advantages. One important motivation is for producing hydrological and associated solute and sediment fluxes. The daily rainfall-runoff model, for example, is a fundamental component of many hydrological analyses, and because of the daily resolution of input rainfall data, these models make no differentiation between n millimetres of rainfall incident on the catchment falling evenly over the period of a day, or in a brief short burst within the day.

Intensity of the rainfall event has an impact on the amount of runoff generated. The shorter the time period over which a rainfall amount falls, or in other words the greater the rainfall intensity, the less incident water will soak into the soil, and the more will run directly into stream channels. As a consequence of the omission of this variability within sample/daily rainfall in the direct runoff ratio in the model structure, daily rainfall runoff models such as IHACRES ([43] and many others) typically under-estimate the runoff generated on days with storms or other intense rainfalls ([27]). It is also well accepted that rainfall intensity affects the mobilisation of sediments and other contaminants. We would
like to address this problem by identifying and tagging intense rainfall events, so that some modification may in the future be made to the runoff calculation (of both water quantity and quality) on these days.

We shall use standard data mining techniques to investigate the nature of extreme rainfall events, utilise k-means clustering (Section 2.3) and Bayesian and decision tree classification methods (Section 2.4). The clustering method was selected primarily for its simplicity and apparent utility, and the classification method on the basis of the wider analysis of Section 4.4. Before proceeding, it is necessary to rigorously define the notion of intensity and non-intensity of a rainfall event. A range of different approaches to this last sub-problem were investigated.

4.2 Training Data

To build and test our intense event extraction method, we utilised data from two areas where rainfall intensity information is available. This is provided by Bureau of Meteorology pluviograph data given at 15 minute intervals (see [2]). Additional daily climate information for investigation of the relation with rainfall intensity was obtained from the Bureau of Meteorology, as collated in the MetAccess database ([67]). Station numbers, names and recording periods used in our analysis are as below:

023000 Adelaide West, 1 June 1955 to 31 August 1977
023031 Adelaide Waite Institute, 1 August 1972 to 31 August 1986
023034 Adelaide Airport, 1 July 1987 to 31 May 1998
040214 Brisbane Regional Office, 1 June 1954 to 30 Jun 1976
040223 Brisbane Airport, 1 July 1986 to 30 April 1998
041044 Hermitage, 1 Feb 1969 to 31 March 1998

The record from Adelaide Waite Institute is supplemented by data from the nearby station, Adelaide Kent Town (023090), to cover missing pluviograph data for the first five years of the record. The first three stations listed above are all in the Adelaide, South Australia area, and the following three located in the Brisbane region of South-East Queensland.
CHAPTER 4. RAINFALL INTENSITY

The choice of stations for analysis was influenced by several factors, the most restrictive being the availability of sufficient periods of high-resolution rainfall data. Just over 1000 stations in Australia collect or have collected this information at some time, although often not for a long period of time. Coupled with the poor quality (by which in this context we mean a high percentage of missing data points) of many of these records, the number of candidate stations is in the low hundreds. From these we deleted those with corresponding daily data that are either poor or absent. Three stations from each of two distant and climatologically distinct locations were chosen from the reduced set of candidates so that similarity and dis-similarity of intensity models within climatic regions and between climatic regions could be examined.

From commonly held expert knowledge ([27]), we can say that the typical high-intensity rainfall events in Brisbane and Adelaide have very different characteristics. Intense rainfalls in Adelaide are likely to be the result of frontal weather systems, whereas those in Brisbane are driven by a more monsoonal weather pattern. Previous work ([142]) has shown that a summer storm type of intense rainfall event is prominent in decision tree intensity models for Brisbane, but the temperate nature of the Adelaide climate makes it unlikely that this type of event will feature significantly in data from this area. The quality of data used in this previous analysis was much higher than we have here, but the number of records much lower.

For each station, with minor exceptions, a daily record consists of the following features (where the abbreviated variable name is also listed if different from the full variable name):

- **Year** (discrete)
- **Day of Year** DayN (discrete)
- **Maximum Temperature** MaxT (continuous)
- **Minimum Temperature** MinT (continuous)
- **Mean Temperature** MeanT (continuous)
- **Rainfall** Rain (continuous)
- **Evaporation** Evap (continuous)
- **Solar Radiation** Rad (continuous)
- **Maximum Wind Speed** Wind (continuous)
4.2. TRAINING DATA

Grass Temperature GrassT (continuous)
Relative Humidity RelHum (continuous)
Sunshine Hours SunHrs (continuous)
Thunder (Boolean)
Hail (Boolean)
Strong Wind StrWind (Boolean)
Gale (Boolean)
Snow (Boolean)
Frost (Boolean)
Fog (Boolean)
Dust (Boolean)
Haze (Boolean)

The discrete Year and Day Of Year variables take integer values, and the last nine variables listed are Boolean presence/absence markers, and exceptions to the database structure are the absence of the variable Solar Radiation at the stations 40214, 41044, and 23000. Strong Winds are defined as those with average speed between 22 and 33 knots, above which the Gale label is applied. All other variables take continuous (to machine precision) real values.

4.2.1 Data Quality

In addition to entirely absent variables, missing points occur throughout most of the dataset. The percentage of missing points in each variable is summarised for all stations in Table 4.1. Date markers and Boolean variables are always present (see Section 1.2.9 for a discussion of this phenomenon), and so are not listed in Table 4.1. The values above are given to zero decimal places, and all variables listed are missing a few points here and there - consider that 36 entries constitute 1% of a record 10 years in length.

Of course, not all variables are relevant. The markers for Snow and Frost, for example, provide no additional information for the Brisbane datasets because snow and frost do not
Table 4.1: Percentage of Records Missing at each Station for Non-Boolean Variables

<table>
<thead>
<tr>
<th>variable</th>
<th>23000</th>
<th>23031</th>
<th>23034</th>
<th>40214</th>
<th>40223</th>
<th>41044</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MinT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MeanT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Rain</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Evap</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Rad</td>
<td>100</td>
<td>1</td>
<td>62</td>
<td>100</td>
<td>53</td>
<td>100</td>
</tr>
<tr>
<td>Wind</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GrassT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>RelHum</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SunHrs</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

occur at any point in the record. The year variable is for the most part eliminated from the analysis, as we assume the weather patterns are stable from year to year.

In Section 4.5.4 later in this chapter, we will attempt to remove variables of limited relevance, reducing the dimensionality of the dataset as far as possible while retaining predictive power. The reasons for this are many: simplicity of human model interpretation, generalisation for use on other datasets that may lack certain variables, elimination of overfitting, and decreased computational complexity. Of these, the first three are true concerns, but the quantity of data we are seeking to process is relatively small and, for most of the methods that we will consider adopting, computation time is not prohibitive.

It is worth noting that for the pluviograph data a ‘day’ is defined as midnight-to-midnight, whereas in the MetAccess database defines a day as 9AM to 9AM. Differently defined days may cause significant difficulties in modelling, as discussed by this author in [145]. A comparison between the rainfall information given in the pluviograph series and the daily rainfall series was made for each station, revealing that summing the pluviograph series over 24 hour periods from 9AM each day does not result in exact correspondence with the daily rainfall series for any dataset. This is perhaps an indication of less than perfect accuracy in the daily rainfall series, but experiments reveal that 9AM to 9AM summation produces a correspondence close to the daily values. In order to reduce the amount of data processing, the daily rainfall series was interpolated to midnight-midnight, rather than summing the pluviograph data. For justification of this approach, see [145], where it is shown that a simple piecewise linear interpolation process adequately transforms a 9AM sampled rainfall to midnight sampled.
4.3 Definition of an Intense Event

The definition of an intense event would depend on the purpose for which intense event information is required. For example, for rainfall-runoff modelling, an intense rainfall event may be a point at which a daily rainfall-runoff model calibrated for the entire record fails to account for the full magnitude of a large runoff peak. However, this approach is both time consuming and may result in some bias towards identifying points at which the model performs badly rather than identifying intense events. The hypothesis that we can usefully identify events of high intensity would thus be bypassed, so here we pursue other possible definitions of an intense rainfall event, obviating the rainfall-runoff modelling of catchments.

In [142], a simple criterion for an intense event was used with good effect, although desirability of a more rigorous definition was noted. The definition was as follows: if on a day in any 15 minute interval the total rainfall exceeds a given threshold (5 mm), then the day is defined as possessing an intense event. Note that 15 minute intervals used were not moving intervals. They were defined as the first, second, third, or fourth quarters of each hour only, as presented in the dataset obtained from [2]. The choice of interval length may have an effect on this measure of intensity, as does the rainfall threshold, but this was not investigated in [142]. The data we have obtained for this analysis are stored in the same format as the previous study.

As we will find, a more sophisticated idea of exactly what we may call an intense event will be required to make analysis more meaningful, especially for the Adelaide data where intense events are less distinct. The need for this is discussed below. In order to extract a more natural definition of intensity, we will look to the data itself for guidance. Throughout the rest of the chapter, an Intense Rainfall Event will now be referred to as an IRE for the sake of brevity.

4.3.1 Clustering Rainfall Events

The search for an appropriate definition of an IRE can be thought of as a quest for a typical event that characterises all IREs. Ideally, other records of intense events would form a cluster around the archetypal event (see Section 2.3) and non-intense events be distant from it. So, if we can isolate a cluster of IREs, whatever characteristics they may have, the cluster centre would provide us with some idea of how an intense event should be defined. This approach has the following advantages over the simple threshold criteria
used in [142]:

- Reduction of arbitrariness in the choice of threshold

- Addresses the issue of variability in IRE characteristics of different climatic regions

- Accounts for whole-event behaviour, rather than short interval behaviour

However, we face a number of obstacles in obtaining an appropriate cluster centre. These obstacles relate to:

- Storage and representation of events

- Choice of cluster regime

- Stability of cluster, under algorithm or parameter changes

We will begin the clustering task by taking the simplest possible approach. This is the ideal starting point for any data mining task, as we shall obtain a benchmark from which to judge subsequent more complex analyses if they are required.

Characterising the rainfall pattern of a given day as an approximate frequency distribution (histogram), we obtain an ordered vector that can be compared to other histogram vectors in a simple Euclidean fashion (see Section 2.6). Note that this transformation results in significant dimensional reduction, and of course a small loss of information. Suitable choice of granularity, ie the number of bins used to generate the histograms, and accounting for missing data points are issues to consider here. As an initial guess, we will examine histograms generated with 10 bins, dropping the dimension of each instance from 96 (24 hours of record sampled at 15 minute intervals) to 10. The bin edges will be linearly defined from zero to 60 mm in increments of 6 mm (per 15 interval). The approximate maximum value of 60 mm was obtained by inspection of the data.

Rainfall events do not always fall wholly within the confines of a 9AM to 9AM day. A more natural definition of the period over which the histogram is calculated may be desirable, based on the beginning and end points of the event. The issue of separating partially indistinct rainfall peaks also arises. However, direct visual inspection of the data reveals that rainfall events rarely cross the 9AM border, and where they do tend to be of very long duration. Therefore, it was decided to use the 9AM to 9AM definition of the day.
4.3. DEFINITION OF AN INTENSE EVENT

Fig. 4.1: Cluster centroids for Brisbane station 040214

Missing points will be ignored, but compensated for by normalising each distribution by the number of points present. This approach was chosen for the sake of simplicity, and as most days with missing sections had only a small portion of the day without records. Note that days with no recorded rainfall are excluded from the clustering process. These points would form the overwhelming majority and it is not necessary to include them at this stage, as they form an intuitively distinct cluster of their own.

Following the theme of simplicity, the histogram clustering scheme we will use is the k-means approach described in Section 2.3. This algorithm requires a priori specification of $k$, the number of clusters. The value $k = 3$ was chosen after experimentation—setting $k > 3$ may result in only three populated clusters regardless. In this way, a stable (random initialisation independent) three cluster structure was extracted from the data for all stations. Without exception, the three generated clusters have the following characteristics: the largest cluster, accounting for 80% to 90% (depending on the dataset) of instances and all the lesser rainfall events; an intermediate cluster consisting of about 12% of instances on average; and the remaining instances belonging to a higher rainfall, higher intensity cluster. In Figure 4.3.1, the three cluster centroids for Brisbane station 040214 are plotted. The distributions of rainfall over each cluster are clearly very different, as we would expect from a good clustering scheme. The behaviour shown in Figure 4.3.1 exemplifies the characteristics of cluster centroids for all datasets, with the exception of 023031, where the profile of the most extreme centroid is almost horizontal, and 041044, where the level of the intermediate centroid rises above that of the higher magnitude centroid at the second sample point.
CHAPTER 4. RAINFALL INTENSITY

Having obtained our clusters, we now select the two clusters of greatest magnitude events, which together account for about 15% of rain days, and label these as extreme \((\text{intensity} = 1)\) to be the first classifier. Now that there exists a training dataset, classification can proceed. Our method is thus a combination of supervised and unsupervised learning (Section 2.3)- the classifier is defined with an unsupervised clustering algorithm, and will be examined with supervised classification techniques. The clustering algorithm provides supervision information to be used in classification.

The classifier built by the method described above was given the label Cnt. A handful of alternative classifiers were also included in the analysis, for purposes of comparison and experiment, these are now described briefly.

4.3.2 Other Classifiers

A variation on the Cnt classifier where all non-intense days (instances) containing one or more missing points are given a missing classifier will be designated Cnt2 (where a missing marker is listed in the dataset, rather than a 0 for non-intense or 1 for intense). The rationale for this experiment is that some of these events may represent an intense event, and feeding the classification algorithm the information that these events are not intense may result in reduced predictive capacity.

We could have chosen to use only the most intense cluster to define Cnt, but this would give us very few intense points on which to train our models, and also probably not account for the number of intense events that may be intuitively expected. Nonetheless, the cluster built by defining only the most extreme cluster as intense will be investigated later as an additional experiment, and shall call this classifier Cnt3. Missing points for Cnt3 are treated as Cnt and Int.

In addition to the Cnt, Cnt2 and Cnt3 classifiers, a classifier generated under the simple threshold criteria mentioned previously and described in [142], called Int, was generated for comparative purposes. Also included is Int2, where missing points as treated as in Cnt2. As an experiment, one more classifier will be tested for the station 040214. This one (Ran) has the same ratio of intense to non-intense events as Int, but the intense label is allocated randomly. This variable is intended as a control, or an indication of the level of apparent information that a classification algorithm will return where there is in fact no structure to be discovered.

Table 4.2 lists the number of events defined as intense out of the total possible (total number of non-missing instances) in the form intense/total, as well as the number of miss-
4.3. **DEFINITION OF AN INTENSE EVENT**

<table>
<thead>
<tr>
<th>model</th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int</td>
<td>143/8128</td>
<td>313/5145</td>
<td>62/3972</td>
<td>578/8214</td>
<td>277/4312</td>
<td>449/10680</td>
</tr>
<tr>
<td>Int2</td>
<td>143/7780</td>
<td>313/4613</td>
<td>62/3552</td>
<td>578/7114</td>
<td>277/3530</td>
<td>449/8646</td>
</tr>
<tr>
<td>Ran</td>
<td>143/8128</td>
<td>313/5145</td>
<td>62/3972</td>
<td>578/8214</td>
<td>277/4312</td>
<td>449/10680</td>
</tr>
<tr>
<td>Missing</td>
<td>205</td>
<td>219</td>
<td>358</td>
<td>522</td>
<td>505</td>
<td>1585</td>
</tr>
<tr>
<td>Cnt</td>
<td>413/7715</td>
<td>374/4771</td>
<td>206/3766</td>
<td>309/7905</td>
<td>191/4121</td>
<td>378/10302</td>
</tr>
<tr>
<td>Cnt2</td>
<td>413/7715</td>
<td>374/4567</td>
<td>206/3426</td>
<td>309/7393</td>
<td>191/3619</td>
<td>378/8735</td>
</tr>
<tr>
<td>Cnt3</td>
<td>47/8081</td>
<td>103/5042</td>
<td>29/3943</td>
<td>27/8187</td>
<td>31/4281</td>
<td>83/10597</td>
</tr>
<tr>
<td>Missing</td>
<td>187</td>
<td>204</td>
<td>340</td>
<td>512</td>
<td>502</td>
<td>1567</td>
</tr>
</tbody>
</table>

Table 4.2: Intense to Non-Intense Count and Missing to Total Count of Instances (Miss1 and Miss2)

...ing intensity markers ('Missing'). Note that the number of missing points is different for the simple criteria Int and the cluster criteria Cnt2 because events satisfying the threshold criteria but having missing points somewhere in the day’s pluviograph record are assigned the intense label regardless of missing patches elsewhere in the day, and is therefore not included in the missing category. Other days containing missing points are listed in Table 4.2 as missing, because it is possible that a complete record would be allocated to either intense or non-intense categories.

Note that there is a strong variation in percentage of days defined as containing an intense event. In general, the Adelaide stations display fewer intense events. Under the 15 minute Int criteria, very few intense events are identified for station 023034 (Adelaide), but this is not the case when the more complex Cnt intense event definition is applied. Number of Cnt IREs is also much greater for Adelaide station 023000, but the reverse is true for all Brisbane stations. The percentage of instances with a 'missing' label are (as expected) very similar for both criteria, and is in no case much greater than 15%.

All of these target variables describe the rainfall intensity at the point at which rainfall and other measurements are taken. At a larger spatial scale, it is possible for an intense event to occur in a catchment and not be identified at the gauging station if the values recorded by that station are not representative of behaviour for the whole catchment area. If rainfall varies significantly over a small spatial scale and an intense event is missed by the measuring equipment, classification algorithms may generate what appears to be a false intensity classification which is in fact correct at a broader spatial scale. We shall proceed using the point measurements discussed above, but noting for further work that some attention to spatial variation of rainfall may be beneficial when generating rainfall...
CHAPTER 4. RAINFALL INTENSITY

intensity variables.

Intuitively, it is appealing to remove all events with no recorded rainfall from the training dataset (possible before clustering histograms), but experiments discussed in Section 4.5.5 show that valuable discriminatory information is contained in these instances. Discarding them results in reduced predictive capacity. Adjustments to the distribution of the intensity variable in training (ie the ratio of intense to non-intense instances included in the dataset) are also made in that section, but they are not found to be useful.

4.4 Classification Algorithms

Several large research projects have focussed on the question of which classification algorithm or family of algorithms is most appropriate for a problem with particular characteristics, for example [115], [113] and [93]. However, for our small problem a trial and error approach, with some expert guidelines, is adequate. Here, we will apply a handful of algorithms and briefly examine the results, before proceeding to optimise and investigate in detail two of these approaches.

In order to obtain the best possible classification models, we must consider the factors that should be considered when choosing an algorithm or algorithms. First consideration is given to accuracy in prediction on an independent dataset. Where possible, we will estimate this using a ten-fold cross-validation scheme (see page 49) rather exclude precious training data from the model learning process by holding out a large fraction for validation.

Second, the highly skewed distribution of intensity markers must be taken into account when considering rates of classification and misclassification - because the overwhelming majority of classifiers are '0' or non-intense, most algorithms will have a tendency to return the majority class. Also, we are starting from the assumption that all events are non-intense (as in a daily rainfall-runoff model). Therefore, every intense event correctly identified represents an increase in knowledge. On the other hand, events incorrectly labelled as intense correspondingly cause a drop in the quality of our knowledge from the base case (no intense events identified). Therefore, the cost of missing an intense event (a false positive) is of little consequence when compared to the incorrect labelling of a non-intense event (a false negative). We have not gained any information over the initial assumption, but we have not lost anything either. The tendency towards majority assignment works in our favour here. For more on the way classifier distribution affects classification, see Section 2.4.6 in the review chapter or [116].
Ideally, the output from the chosen classification algorithm should be physically interpretable to a human observer, but this is not necessary from a machine learning perspective. Future use of the model, ie prediction on unclassified data, will be performed by a computer that cares nothing for physical interpretability, so we will not exclude algorithms that do not satisfy this criteria. It is, however, desirable. Computational complexity is also an issue to consider, although dataset size is relatively small for our case study. There exists a gargantuan library of classification algorithms readily available, so we will select a few from different families to compare. The following algorithms from the Waikato Environment for Knowledge Analysis (WEKA) library, which is one of the most trusted and widely applied data mining packages available, were tried:

- OneR
- Naïve Bayes
- Naïve Bayes with kernel estimator
- C4.5
- ADtree
- Logistic Regression
- Neural Net (feedforward)
- Voting Feature Intervals (VFI)

In addition to a high level of functionality, the WEKA package is open source and distributed under GPL. The wealth of documented projects using WEKA and freely available source code inspire confidence in the correctness of implementation of the algorithms in this piece of software.

The Naïve Bayes method is described in Section 2.4. Naïve Bayes with an additional kernel estimator was also applied. The ADtree algorithm, an alternating decision tree extractor, are also covered briefly in that section. Quinlan’s popular C4.5 algorithm ([117]) described in Section 2.4.2 and implemented as J48 in Weka. The Logistic Regression and Neural Net algorithms are exactly as named, and OneR (see Section 2.4.7) is a simple scheme utilising multiple splits over one variable only. The Voting Feature Intervals method is described in [32]. The use of nearest-neighbour techniques were considered, but such
CHAPTER 4. RAINFALL INTENSITY

methods do not produce a succinct model for further use (see Section 2.4.8). In addition, as several data types exist within the dataset (ordered categorical, Boolean, real) the application of a similarity measure presents additional difficulties. For these two reasons the concept was abandoned.

Station 023000 from Brisbane and 040214 from Adelaide were selected arbitrarily to test the various algorithms. Model performance was quantified by confusion matrices generated by averaging the results of a ten-fold cross-validation procedure (see Section 2.4.9). The only exception to this rule was the Neural Network algorithm (a simple feedforward implementation), for which cross-validation proved time-prohibitive. In this case the model was constructed using 66% of the dataset and tested on the remainder.

The confusion matrix concept was introduced in Section 2.4.9. Consider the confusion matrix:

\[
\begin{bmatrix}
A & a \\
\bar{b} & B
\end{bmatrix}
\]

Recall that A is the number of null (0 or non-intense) instances correctly classified, \(a\) is the number of null instances classified as intense (false positive). Similarly, B is the number of correctly classified intense days and \(b\) the count of intense instances classified as null (false negative). Therefore, \(A+a\) is the number of non-intense instances and \(B+b\) the number of intense rainfall events in the record. An optimal confusion matrix would be purely diagonal.

For OneR, several runs were performed with varying parameters. The minimum number of instances covered by every rule of OneR was varied through three, six and twelve. Most of the algorithms above have tunable parameters of some sort, but as time considerations prevent rigorous optimisation by repeat trial of every algorithm, we will use parameter values that are default or appear reasonable for our class of problem, and only perform a few investigatory experiments. Indicative results are reported in Table 4.3. In Section 4.5, a more detailed investigation into parameter values is performed.

It is may be thought that the distribution of the training data should match the distribution of the true real-world distribution of the classifier, but recent work such as [155] finds that this is not necessarily true. In practice, the sample distribution is often deliberately altered in the knowledge discovery process. If the distribution of the classifier is highly skewed, classification models can be built in a cost-sensitive fashion, to ameliorate the tendency towards prediction of the most populous class. The distribution of values in the classifier (in this case the cluster-based intensity measure Cnt2) is highly uneven,
in that only about 10% of instances are intense. However, the cost of false negatives and false positives are not equal, as discussed above, so we will not attempt to weight misclassifications in this way.

Table 4.3 details the confusion matrices generated by all algorithms. The OneR algorithms are given a separate comment towards the end of this section. As the simplest of models, excepting only the allocation of all classifier values to the mode, they provide us with a benchmark level of classification below which an algorithm can be dismissed as inappropriate- if a complex model cannot exceed the performance of a simple model, the simple model is clearly superior.

When reading confusion matrices, note that the top and bottom lines of the matrix will always sum to the total number of non-intense and intense event respectively, for a given classifier. Cnt, Int, and the other classifiers will have different row sum because the number of intense events identified by each scheme varies.

The C4.5 algorithm and the ADtree scheme generate decision trees of slightly different character but equal interpretability. Table 4.3 demonstrates that both of these methods perform well. In all cases, the number of correctly identified intense events is high (greater than 50%) and the number of false positives very low in comparison to most other methods (less than 2% of non-intense event are labelled as intense). ADtree is slightly more expensive in terms of computation time, although even with 10-fold cross-validation the difference is not significant. For most of the following analysis, we will use these two methods, although other algorithms will be used for specific tasks such as feature selection.

Note that using the Naive Bayes classification algorithm, the error rate of the minority (intense) class is less than that of the majority (non-intense) class. This contravenes the
common trend that minority error rates should be the greater, as explained in Section 2.4.6, and the Naïve Bayes is the only algorithm that produces this result. Voting Feature Intervals (VFI) tends to correctly identify more intense events than the decision trees, but also has a larger rate of instances incorrectly labelled intense. Logistic Regression identifies few intense events, as does the neural network approach. It is possible that neural nets could be made to perform better with more tuning, but as available time is limited and this approach would require considerable investment in labour, the option was not explored.

The OneR results are comparable with if not quite as good as the more complex methods. The classifier OneR (see Section 2.4.7) uses multiple splits over one feature only. This gives us some impression of the amount of information contained within the most important feature without reference to the other descriptive attributes. This is of course Rain2, the daily rainfall. From Table 4.3, we see that the OneR algorithm with a bucket size of 3 can extract about 95% correct classifications from Rain2. Any gains we have made by introducing other variables into the learning environment with a more complex scheme such as ADtree or C4.5 are as yet restricted to an additional 50% true positives and a reduction of about 40% in false positives. Taking for example station 023000, where OneR (b=12) identifies roughly half the intense events with 141 false positives. For the same station, both C4.5 and ADtree identify only an additional 13% of the total number of intense events, although both methods result fewer false positives than OneR in all cases.

4.5 Results of Classification using C4.5 and ADtree

In this section we will consider the behaviour of the two classification schemes ADtree and C4.5 when applied to all six stations in the study, with optimisation over appropriate parameters (these are listed in Section 4.5.3). The aim is to make as much improvement over the results listed above as possible. Various classifiers (ie differing definitions of an intense event) will also be tested. In addition to examination of confusion matrices, issues such as model complexity and physicality will be briefly addressed. A later subsection will detail attempts to reduce model dimensionality by eliminating variables not contributing usefully to the analysis.

4.5.1 Results of Cnt2 Classification

Cnt2 which, as discussed above, defines an intense event as a day falling into either of the two higher magnitude clusters and includes provision for missing data, is the base case
4.5. RESULTS OF CLASSIFICATION USING C4.5 AND ADTREE

<table>
<thead>
<tr>
<th>model</th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>7399 129 4479 88 3366 60 7309 84 3555 64 8611 124</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>182 231 255 119 106 100 101 208 102 89 219 159</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADtree</td>
<td>7419 109 4460 107 3367 59 7304 89 3568 51 8635 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>178 235 241 133 114 92 93 216 81 110 196 182</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Cnt2 Confusion Matrices, C4.5 and ADtree Algorithms

classifier to which we will compare others. First, let us consider the confusion matrices obtained by application of the ADtree and C4.5 algorithms with ten-fold cross-validation for each dataset. These are given in Table 4.4.

A number of preliminary observations can be made. First, performance levels are better for Brisbane datasets, which are those prefixed with 04, rather than the Adelaide stations which are prefixed by 02. For example, the average percentage of correctly identified IREs is 45 for Adelaide and 55 for Brisbane, and the average percentage of false positives (to all non-intense events) is 1.8 for Adelaide and 1.3 for Brisbane. By far the best performance is achieved with data from station 040214, where almost 70% of IREs are correctly identified. The average performance measures for the other two Brisbane stations are closer to the Adelaide average.

This superior performance is unlikely to be due to higher quality daily data, as we have seen in Table 4.1 that levels of missing records are similar, and all stations were selected from the list of pluviograph stations as having the fewest possible missing records. Each one was collected at an apparently rigorous source - airports, research institutes, and Bureau of Meteorology regional offices. These were preferred over post offices or similar arrangements. However, it is true that the proportion of missing days is not a definite indicator of data quality. Uncertainty and measurement error must also be considered. In fact, we know from Section 4.2.1 that rainfall totals are subject to uncertainty (as is all data, to varying degrees).

Brisbane datasets seem to be slightly better suited to analysis with the ADtree algorithm, but the margins are very small. The number of true positives identified by C4.5 is less than that by ADtree by 3 to 10% of the total number of IREs, while the number of false positives rises only in one case by a very small amount. We can see that the confusion matrices for Adelaide stations (04**) do not vary much between algorithms. In the following subsections, options regarding classifier choice and parameter values are explored.
CHAPTER 4. RAINFALL INTENSITY

Table 4.5: Int Confusion Matrices

<table>
<thead>
<tr>
<th>model</th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>7747</td>
<td>33</td>
<td>4609</td>
<td>4</td>
<td>3536</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>116</td>
<td>27</td>
<td>310</td>
<td>3</td>
<td>49</td>
<td>13</td>
</tr>
<tr>
<td>ADtree</td>
<td>7770</td>
<td>10</td>
<td>4612</td>
<td>1</td>
<td>3534</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>23</td>
<td>313</td>
<td>0</td>
<td>57</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.6: Cnt Confusion Matrices

4.5.2 Results of Int, Cnt, and Cnt3 Classifications

For the purposes of comparison, we will give here the confusion matrices for some differently defined criteria for intensity.

Of the many conclusions we may draw from the classification rate information summarised in Tables 4.4, 4.5, 4.6, and 4.7, we first note that the introduction of the missing value marker increases performance. That is, the confusion matrices for Cnt2 are generally better than those for Cnt. Note however that the concept 'better' is somewhat subjective, as we must consider not only how close each matrix is to pure diagonality, but also the different costs of type I and type II errors. This is as expected, because we are removing what may be false information at the cost of removing a relatively small amount of true information. For Adelaide stations, note that the number (and fraction) of correctly identified IREs is very low for Cnt3 and Int classifiers. For Cnt3, this problem also appears in Brisbane station 041044 and may be due in part to the fact that only a very small fraction of instances are assigned the intense marker. If Cnt3 was to be used in further analysis, some resampling of the dataset may be used to counter this problem. However, for station

Table 4.7: Cnt3 Confusion Matrices

<table>
<thead>
<tr>
<th>model</th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>8067</td>
<td>14</td>
<td>5042</td>
<td>0</td>
<td>3934</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>15</td>
<td>103</td>
<td>0</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>ADtree</td>
<td>8064</td>
<td>17</td>
<td>5042</td>
<td>0</td>
<td>3940</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>20</td>
<td>103</td>
<td>0</td>
<td>25</td>
<td>4</td>
</tr>
</tbody>
</table>
040223 more than half Cnt3 IREs can be recovered.

The correct classification rates for Int, the simple 15-minute threshold criteria, are considerably lower than those for Cnt2 (note that missing instances are marked in Int as well as in Cnt2) in almost all cases. Superficial examination of Tables 4.5 shows that the number of Adelaide true positives is very poor. Int Brisbane true positive rates are closer to the those achieved by Cnt2. Cnt Brisbane and Adelaide true positive rates are comparable with the Cnt2 results, and indeed may be slightly greater. However, the false positive rate is in many cases also greater. The average false positive rate remains constant between the two cases.

The results for Cnt and Cnt2 classifiers are broadly very similar. We have seen in this section that the events marked by Cnt and Cnt2 are in general more easily identifiable than those defined by Int, and that the addition of a missing variable marker has little affect on performance of the classifier generated by clustering. Therefore, we shall proceed to optimise the C4.5 and ADtree algorithms for the Cnt2 classifier knowing that this is as good as or better than any other classifier for this task.

4.5.3 Parameter Selection for C4.5 and ADtree Algorithms

The software default parameters for the C4.5 Weka implementation of C4.5 are as follows:

- **binarySplits = False** Sets binary splitting for nominal attributes. As all of our nominal attributes are binary anyway, this option does not concern us.

- **confidenceFactor = 0.25** Minimum confidence (see page 52) level, for pruning branches.

- **minNumObj = 2** Minimum number of instances to be covered by a leaf or decision node

- **numFolds = 3** For reduced error pruning.

- **reducedErrorPruning = False** Switch for reduced error pruning.

- **saveInstanceData = False** A memory management option.

- **subtreeRaising = True** Switch for subtree raising.

- **unpruned = False** Switches pruning on and off.

- **useLaplace = False** Laplace Smoothing for predicted probabilities. We will not use this option.
Subtree Raising is the replacement of a branch with a sub-branch of that branch, under certain conditions. The Reduced Error Pruning option enables the algorithm to hold back a part of the training set from the building of the tree and call this the pruning set. The unpruned tree is then tested on this set, and pruned accordingly before the final evaluation on the testing set occurs. The inverse of the numFolds parameter is the fraction of training set used as the pruning set. A brief description of these concepts is provided in [11]. Note that the Reduced Error Pruning option automatically eliminates Subtree Raising. For the ADtree algorithm, we have the following options, with defaults given.

\textbf{numOfBoostingIterations} = 10 See Boosting (Section 2.4.5).

\textbf{randomSeed} = 0 Random number seed, for initialisation.

\textbf{saveInstanceData} = False A memory option.

\textbf{searchPath} = Expand all paths The other options available here are Expand heaviest path, Expand best z-pure path, and Expand random paths.

The default choice for the searchPath parameter provides a comprehensive search for optimal structure, while the others allow choice between possibilities randomly or by weighting according to an appropriate measure (see Section 2.4.2).

Here, the greatest potential for model improvement exists with numOfBoostingIterations, although for the sake of interest we will experiment a little with searchPath and randomSeed as well. Tables 4.8 and 4.9 give confusion matrices for alterations from default options for the C4.5 and Adtree algorithms. Abbreviations are as follows: Base implies the

<table>
<thead>
<tr>
<th>station</th>
<th>Base</th>
<th>REP=T</th>
<th>StR=F</th>
<th>Unpruned=T</th>
</tr>
</thead>
<tbody>
<tr>
<td>023000</td>
<td>7399</td>
<td>129</td>
<td>7399</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>182</td>
<td>231</td>
<td>181</td>
<td>232</td>
</tr>
<tr>
<td>040214</td>
<td>7309</td>
<td>84</td>
<td>7316</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>208</td>
<td>87</td>
<td>222</td>
</tr>
</tbody>
</table>

![Confusion Matrix](image)

Table 4.8: Parameter-Varied Confusion Matrices, C4.5
4.5. RESULTS OF CLASSIFICATION USING C4.5 AND ADTREE

<table>
<thead>
<tr>
<th>station</th>
<th>Base</th>
<th>nBI=20</th>
<th>nBI=5</th>
<th>Heaviest</th>
<th>Z-pure</th>
</tr>
</thead>
<tbody>
<tr>
<td>023000</td>
<td>7419</td>
<td>7411</td>
<td>7330</td>
<td>7410</td>
<td>7410</td>
</tr>
<tr>
<td></td>
<td>178</td>
<td>168</td>
<td>118</td>
<td>177</td>
<td>177</td>
</tr>
<tr>
<td>040214</td>
<td>7304</td>
<td>7300</td>
<td>7274</td>
<td>7308</td>
<td>7308</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>93</td>
<td>82</td>
<td>95</td>
<td>95</td>
</tr>
</tbody>
</table>

Table 4.9: Parameter Varied Confusion Matrices, ADtree

<table>
<thead>
<tr>
<th></th>
<th>023031</th>
<th>023034</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>023000</td>
<td>4480</td>
<td>3334</td>
<td>56</td>
<td>8628</td>
</tr>
<tr>
<td>040214</td>
<td>246</td>
<td>89</td>
<td>92</td>
<td>187</td>
</tr>
</tbody>
</table>

Table 4.10: Other Stations with Reduced Error Pruning, C4.5

default case, REP Reduced Error Pruning, StR Subtree Raising, Con Confidence, and minNO minNumObj for C4.5, nBI numBoostingIterations, and Heaviest and Z-pure the path expansion options. Varying the initialisation parameter randSeed has no effect for our problem.

As with the initial experiments with multiple algorithms, stations 023000 and 040214 were used to explore optimisation options. Clearly, the only significant improvement that we have encountered occurs with the use of reduced error pruning of the C4.5 decision tree. Applying this to the other four datasets, the confusion matrices listed in Table 4.10 are generated.

All of these represent small gains over the base case (compare with Table 4.4). Also, it should be noted that the trees generated with Reduced Error Pruning are typically much smaller (less complex) than the base trees (for example, the tree size for C4.5 + REP on station 023000 data is 45 nodes, where the default C4.5 tree contains 135 nodes, and is the largest tree for any station). The additional computation time required by this method is negligible when compared to the savings generated by the reduced model complexity—no appreciative difference in computation time was noted. Therefore, we will adopt this option in the remaining experiments.

None of the ADtree options provide worthwhile gains over the base performance level, but the alternate path expansions yield a reduction in computation time. While the ADtree algorithm has a much higher computational cost than the C4.5 implementation, the total amount of time taken by the computation still very small, so we will remain with the comprehensive search option.
4.5.4 Attribute Selection

Initially, each of our six datasets involved 20 or 21 variables in any particular classification problem. Those with 20 are missing Solar Radiation records. Ideally, we would like to reduce this number as far as possible without sacrificing significant predictive capacity, in order to obtain as compact a model as possible. The benchmark OneR classification results discussed in Section 4.4 are not very distant from the results obtained above. This implies that the level of information contained in the foremost attribute is not hugely less than the total (rainfall intensity) information content, so attribute removal is a very viable option. It is sometimes even the case that removing extraneous features can actively improve predictive performance at the modelling stage, for example see [57]. Of course, the decision tree induction process as described in Section 2.4.2 should ideally utilise only relevant features, and does not account for redundancy between variables. We must assume that there is, and look for, some redundancy in our climate dataset, where (for example) we have four different daily measures of temperature. Some of these can almost certainly be removed without loss of significant information.

From an intuitive perspective, we should expect that certain variables provide little or no useful contribution to the information content of the dataset, such as the Booleans Snow and Frost which occur very sparsely. For sub-tropical Brisbane, they provide no information at all. Weka provides automated tools for dimensionality reduction, and we will use one of these to extract a subset of the features with which to classify. Also, we will try some combinations that seem intuitively appealing, like the removal of all but one Temperature variable.

There are many options for attribute selection methods. Our choice of automated algorithm will be the greedy hillclimber BestFirst (see [112]), the usual search strategy for wrappers ([77], [57]), combined with a facility for evaluating the appropriateness of candidate subsets using a classifier (ClassifierSubsetEval). The standard C4.5 algorithm will be our classifier. Using these options, Table 4.11 below summarises the presence of each feature in the variable subsets we obtain, for each of the six study datasets.

Using these feature subsets, the resulting models yield the confusion matrices in Table 4.12. In all datasets excepting 041044, these matrices represent a small drop in performance below the base case (C4.5 + REP and ADtree). For station 041044, the performance improves slightly. All changes are of very small order, almost negligible. Adding Reduced Error Pruning to the C4.5 classification yields subsets as given by Table 4.13. It may be noted that these are subsets of the selected feature subsets given above in Table 4.11.
### Table 4.11: Summary of Dimensionally Reduced Subsets

<table>
<thead>
<tr>
<th>Feature</th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>DayN</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>MaxT</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>MinT</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>MeanT</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rain2</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Evap</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Rad</td>
<td>n/a</td>
<td>yes</td>
<td>no</td>
<td>n/a</td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>Wind</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>GrassT</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>RelHum</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>SunHrs</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Thunder</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Hail</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>StrWind</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Gale</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Snow</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Frost</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Fog</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Dust</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Haze</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

### Table 4.12: Confusion Matrices from Selected Feature Subsets, C4.5 + REP and ADtree

<table>
<thead>
<tr>
<th>Feature</th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>7415</td>
<td>113</td>
<td>4490</td>
<td>77</td>
<td>3368</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>184</td>
<td>229</td>
<td>256</td>
<td>118</td>
<td>112</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>7310</td>
<td>83</td>
<td>87</td>
<td>222</td>
<td>106</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>3561</td>
<td>58</td>
<td>8607</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>140</td>
<td>88</td>
<td>221</td>
<td>207</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112</td>
<td>88</td>
<td>221</td>
<td>202</td>
<td>176</td>
<td></td>
</tr>
<tr>
<td>ADtree</td>
<td>7419</td>
<td>109</td>
<td>4455</td>
<td>112</td>
<td>3368</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>178</td>
<td>235</td>
<td>234</td>
<td>140</td>
<td>116</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>7302</td>
<td>91</td>
<td>88</td>
<td>221</td>
<td>80</td>
<td>111</td>
</tr>
<tr>
<td></td>
<td>3568</td>
<td>51</td>
<td>8638</td>
<td>97</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>202</td>
<td>176</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>023000</td>
<td>023031</td>
<td>023034</td>
<td>040214</td>
<td>040223</td>
<td>041044</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>DayN</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>MaxT</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>MinT</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>MeanT</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Rain2</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Evap</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Rad</td>
<td>n/a</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>Wind</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>GrassT</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RelHum</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SunHrs</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Thunder</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Hail</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>StrWind</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Gale</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Snow</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Frost</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Fog</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Dust</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Haze</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 4.13: Summary of Dimensionally Reduced Subsets, with REP
It is worth noting the differing variables that are likely to be selected in Adelaide and in Brisbane. For example, *Thunder* shows no sign of selection in Adelaide, but occurs in every Brisbane subset. *Hail* is a far more popular indicator of intensity in Adelaide (possibly because of the rarity of hail in the Brisbane record), and relative humidity applies more to Brisbane stations.

In general the confusion matrices resulting from this analysis are slightly better than those obtained with C4.5 and default options, but are comparable to the tuned C4.5 models of Table 4.10. A strong trade off exists—where one entry of the confusion matrix measure improves, another becomes poorer. For example, both the number of IREs correctly identified for station 023034 with a reduced feature set both rise slightly, where these values fall for station 041044. The attribute-selected trees are, somewhat counter-intuitively, larger.

C4.5 confusion matrices for selection with Reduced Error Pruning (REP) are listed in Table 4.14 below and a comparison of tree sizes with and without REP in Table 4.15.

A similar analysis was performed using the BestFirst algorithm with the ADtree classifier. The performance in classification of the resulting feature subsets is on the whole similar to that listed above. The contents of the feature subsets themselves are quite similar, excepting that those generated with C4.5 classification are in every case at least as great in terms of number of features as the ADtree counterpart and often more so. The largest subset generated with Adtree contained five predictor features. For example, for station 023031, dimensions reduce to *Rain2* only with ADtree, whereas with C4.5 the subset consists of *Rain2* and *SunHrs*. Constrained by the structure of the ADtree algorithm, all tree sizes for this experiment were 31. It should also be noted that feature selection with ADtree is computationally expensive—the small increase in cost noted in Section 4.4 becomes considerable when many applications are made.

For comparison with the feature subsets selected by automated means and resulting
classification models, we will now consider which a set of variables we might intuitively expect to best represent the rainfall intensity in all datasets. There are two obvious courses to take: the removal of certain Boolean variables, especially Snow, Frost, and Dust; and the reduction of the temperature variables.

Note from Table 4.13 that most of the Booleans are not selected for the majority of datasets. Frost, Dust and Snow are never used (we would expect this, as precious few recordings of presence exist). The only Boolean that is selected (three times out of six) is Thunder, which we will keep because of the assumed correlation between presence of thunder and storm events. In keeping with this strategy, Rain2, Wind, GrassT, RelHum, and SunHrs are retained due to their selection three or more times out of six. This leaves six predictor variables in the feature subset, which we will call subset 2.

Tree sizes for C4.5 in Table 4.17. ADtree tree sizes are constant at 31 nodes. In this case the tree sizes are in some cases smaller than the best case- not all of the selected variables appear in the tree.

One more feature subset experiment was performed. This time, we considered only the known physical properties of the system and the data. First, we shall remove all the Booleans except Thunder, as they are questionable in quality and typically very sparse. From the temperature measures, we will remove MeanT and MinT, leaving us with MaxT and the (generally) most extreme minimum temperature measure GrassT. Considering that we may estimate evaporation from sunshine hours and temperature measures (especially grass temperature, and that evaporation will be strictly increasing in both these measures), Evap is eliminated. Rad can be removed by similar consideration of sunshine hours and temperature. It is also the variable with the least complete record.

<table>
<thead>
<tr>
<th></th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>7409</td>
<td>119</td>
<td>4473</td>
<td>94</td>
<td>3337</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>173</td>
<td>240</td>
<td>249</td>
<td>125</td>
<td>82</td>
<td>124</td>
</tr>
</tbody>
</table>

Table 4.16: Confusion Matrices Resulting from Feature Subset 2

<table>
<thead>
<tr>
<th></th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5 (REP+sel)</td>
<td>15</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>C4.5 (base case)</td>
<td>135</td>
<td>31</td>
<td>93</td>
<td>95</td>
<td>41</td>
<td>71</td>
</tr>
</tbody>
</table>

Table 4.17: Tree Sizes, Feature Subset 2
4.5. RESULTS OF CLASSIFICATION USING C4.5 AND ADTREE

Table 4.18: Confusion Matrices Resulting from Manually Selected Feature Subsets

<table>
<thead>
<tr>
<th></th>
<th>023000</th>
<th>023031</th>
<th>023034</th>
<th>040214</th>
<th>040223</th>
<th>041044</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>7413</td>
<td>183</td>
<td>4487</td>
<td>3345</td>
<td>7313</td>
<td>3570</td>
</tr>
<tr>
<td></td>
<td>115</td>
<td>230</td>
<td>80</td>
<td>81</td>
<td>80</td>
<td>49</td>
</tr>
<tr>
<td>ADtree</td>
<td>7423</td>
<td>188</td>
<td>4455</td>
<td>3372</td>
<td>7298</td>
<td>3563</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>225</td>
<td>112</td>
<td>54</td>
<td>95</td>
<td>56</td>
</tr>
</tbody>
</table>

This leaves eight features: \textit{DayN}, \textit{MaxT}, \textit{Rain2}, \textit{Wind}, \textit{GrassT}, \textit{RelHum}, \textit{SunHrs}, and \textit{Thunder}. Confusion matrices for C4.5 and ADtree schemes are given in Table 4.18 below.

Comparison of Tables 4.16 and 4.18 reveal that neither scheme gives better performance than the other for all datasets, and in most cases the confusion matrices are of comparable quality. Both of these intuitive schemes exhibit a slight tendency towards increasing the true positive rate at the expense of false positives when compared to the base case. The automated attribute selection algorithm performs better in general than either of these, with results comparable to the base case and a reduction in number of variables from 20 to six to eight. Also, the tree sizes generated from all three dimensionally reduced datasets are comparable - of the order of 30 nodes. Therefore, we conclude that the automated parameter selection method is sound. However, we must note that the number of nodes in final trees are typically much larger for the automatically generated subsets.

### 4.5.5 Additional Experiments

As well as the attribute selection and parameter variation experiments discussed above, a few other avenues were explored. These options include randomisation on the target data, metaclassification by partitioning the dataset, and altering the distribution of the classifier, and we report on these below.

#### Randomisation

A series of associated experiments were performed on the example datasets 040214 and 023000. The first of these involved the randomisation of the Cnt2 classifier (distribution of 1, 0, and unknown value markers preserved), which was then used to generate C4.5 and ADtree classification trees. The object of this exercise was to examine the degree of information the algorithms seem to be extracting from a dataset with the same characteristics as those used in this study, but where no causal relationship exists between the classifier
and other features in the instance. In all cases, the classification schemes simply predicted the majority classifier, as we would hope. The size of the generated C4.5 trees were both unity, but the ADtree algorithm produced trees of size comparable to that generated from the non-randomised datasets, due to the nature of the algorithm.

**Metaclassification**

In the early stage of experimenting with possible algorithms, it was noted that the Naive Bayes algorithm typically has a high true positive rate (identifies intense events) better than the C4.5 or ADtree algorithms, but a false positive rate (non-intense events labelled as intense) that is prohibitively high. Upon examination of final results obtained with Cnt2 and C4.5/ADtree it was decided to investigate the possibility of using a metaclassification approach (see page 39). This approach combines the predictions of multiple classifier models— for example, if model A predicts well in some region and classifier B performs well in another, a metamodel that considers classifier A on some occasions and classifier B on others may be appropriate. See for example [115] on metalearning.

However, preliminary examination of the correct/incorrect classifications of Naive Bayes, C4.5, and ADtree algorithms implied that all three classification schemes perform well in the same regions of the input dataset (where daily rainfall is very small or very large, for example), and poorly in the same regions (eg medium rainfalls). Therefore this option was abandoned.

**Distribution Adjustment**

As noted in Section 4.4, the distribution of intense to non-intense days is highly uneven, IREs making up around 10% of each record. Where false positives and false negatives are of equal importance, the ideal situation would intuitively be equal numbers of each classifier (one and zero). When the distribution is skewed, some re-sampling of the dataset may be justified. However, as we wish to minimise false positives or type I errors, we have hereto left the highly skewed distribution in place.

After [117], consider the following example: the simplest possible classifier for a problem with 100 instances, 90 of which belong to one class, would assign every instance to the majority class, and in this case report an accuracy of 90% and the confusion matrix below.

\[
\begin{bmatrix}
90 & 0 \\
10 & 0 \\
\end{bmatrix}
\]
4.5. RESULTS OF CLASSIFICATION USING C4.5 AND ADTREE

Table 4.19: Confusion Matrices, Adjusted Distribution

<table>
<thead>
<tr>
<th></th>
<th>C4.5</th>
<th>C4.5+REP</th>
<th>Bayes</th>
<th>ADtree</th>
</tr>
</thead>
<tbody>
<tr>
<td>023000</td>
<td>7292</td>
<td>236</td>
<td>7197</td>
<td>331</td>
</tr>
<tr>
<td>0</td>
<td>7434</td>
<td>0</td>
<td>7434</td>
<td>0</td>
</tr>
<tr>
<td>040214</td>
<td>7250</td>
<td>143</td>
<td>7729</td>
<td>164</td>
</tr>
<tr>
<td>0</td>
<td>7416</td>
<td>0</td>
<td>7416</td>
<td>0</td>
</tr>
</tbody>
</table>

On the other hand, if we have a classifier that assigns instances to the majority class with a probability of \( p = 0.9 \) (reflecting the class distribution), the expected confusion matrix would look like this.

\[
\begin{bmatrix}
    p \times p & p(1-p) \\
    (1-p)p & (1-p)(1-p)
\end{bmatrix} \times 100 = \begin{bmatrix}
    81 & 9 \\
    9 & 1
\end{bmatrix}
\]

Note that the second classifier would have an accuracy of 82%, which is less than the accuracy of the majority classifier. We must keep this in mind when considering the amount of information delivered by the various classification schemes. This example also underscores the value of confusion matrices in performance quantification over a single percentage correct classification rate.

As an experiment, we now resample the test stations 023000 and 040214 in a common way—by replicating the intense instances an appropriate number of times to approximately match the number of non-intense instances. Where distributional adjustment is performed by this method, the modeller must also be careful to avoid overfitting. For example, a leaf may appear to cover 12 instances, but these may have been generated by four instances replicated three times. Applying C4.5 and C4.5 + REP, Naïve Bayes, and ADtree algorithms, we obtain the confusion matrices as in Table 4.19.

Interestingly, the poor performance of the Naïve Bayes algorithm, which was attributed to distribution problems in the initial survey of methods, still shows the characteristic high level of false positives although it is also true that all of the other algorithms also exhibit this tendency when compared to the C4.5 + REP or ADtree standard cases, with or without attribute selection. For this reason, we will abandon this avenue of inquiry.

Dry Day Removal

Another approach to the distribution problem above is to remove some non-intense days. The obvious course of action is to remove instances with zero rainfall from the record
altogether. This brings the number of zero classifiers down to at least the order of the unity classifiers. Confusion matrices subsequently obtained for the test datasets are given in Table 4.20. It appears that the predictive capacity of the data is reduced by the removal of dry days. Thus, we can conclude that important information is contained in these days, which helps the classifier distinguish between the characteristics of intense and non-intense days.

### 4.5.6 Regionalisation

Recall that the datasets fall into two groups of three: Brisbane and Adelaide. The motivation behind selecting multiple stations in one geographic region was twofold - to examine the commonality of trees and rules for the same and for different climatic schemes, and thereby to test the possibility of regionalisation of the classification models. Regionalisation refers to the commonality of our models across stations within a region, and from region to region.

The structure of decision tree algorithms and of our particular problem do not allow us to isolate unique and optimal decision trees for each station. Therefore, rather than compare the structure of trees themselves, our approach will be simply to apply the optimal models for each dataset to the data from other stations. We would, of course, still like the trees to be as easily interpretable (small) as possible, so we will apply C4.5 using the REP option. The confusion matrices output from this process are given in Table 4.21. In the first column, the last two digits of the dataset generating the model are given.

We can divide these results into four groups. The first consists of the upper left block, where Adelaide models are applied to Adelaide data. In the second block on the upper right these models are applied to Brisbane datasets, the third block at bottom left reports results of Brisbane models are applied to Adelaide data, and the fourth and last block where Brisbane models are applied to Brisbane station data.

In the first and fourth blocks, the quality of result is in general comparable with the
results listed in Table 4.10 and under the appropriate REP option in Table 4.8. This implies that within both the Brisbane and Adelaide regions, the models regionalise well. Let us consider station 023034, where the difference between usually reported confusion matrices are as notable as in any other case. Here, the number of correctly identified IREs falls from 117 in Table 4.10 to 109 where the model generated with data from station 023031 is applied in Table 4.21, but the number of false positives also falls. These results still imply a useful model. When the station 023000 model is applied to this data, the same effect occurs more strongly - while only one third of IREs are identified, the false positive rate is very small indeed.

Examination of blocks two and three reveals that block two (Adelaide models) have a very high false positive rate, and block three (Brisbane models) show a strong tendency to predict the majority class, resulting in a high false positive rate. We can conclude from this that regionalisation across regions from Brisbane to Adelaide or vice-versa is not appropriate. This should not surprise us overmuch, due to the very different climatic characteristics of these two regions. Following our example station 023034 to block three, where Brisbane models are applied, we see that the classifier is approaching the case where the majority classifier (not intense) is assigned in all cases. Thus, very few Adelaide events meet the identified intensity requirements for Brisbane data - this is not surprising, as Brisbane is a much wetter system. In block two, applying the model generated by station 023034 to Brisbane station reveals that many events are identified as intense, and type I errors are high. Many events that have not been defined as intense using Brisbane criteria.
fit the requirements for Adelaide intense events. It is notable in this block that a small number of IREs have not been labelled intense, and therefore, processes other than simple magnitude are at work.

4.6 Conclusions

In the course of this chapter we have seen that the information contained in the variable Rain2 is sufficient to explain about 50% of intense rainfall events with a loss of only a few percent of true non-intense days. Our efforts to increase the reliable classification of IREs above the 50% mark without incurring significant penalties were most successful using the C4.5 algorithm with Reduced Error Pruning (REP), gaining us of the order 20% extra true positives. The ADtree algorithm also produced good results.

In the case of station 023031, our models consistently performed more poorly in most experiments than all the other datasets. Recalling Section 4.2, we find that this record was the only composite, a missing block of pluviograph data being adopted from the nearby station 023000 Adelaide Kent Town. This may be the reason for poor performance, but if so, the implications on the geographical sensitivity of the daily variable data are surprisingly significant. It would be easy to assume that geographically close station would produce similar rainfall records, but the poor performance of station 023031 sheds doubt on this. Regionalisation of the intensity models themselves seems to be very viable within local geographical regions, but not across different climatic schemes.

In general, Brisbane stations performed better than Adelaide. This was expected, as the extreme events in the Brisbane region are more likely to be characteristic summer storm events, rather than the more elusive frontal events more common to the Adelaide area. The monsoonal summer storms are well characterised by high temperature, location in the summer months, and high humidity. Frontal events have more varying characteristics.

We may consider another division in the datasets, by number of elements - 023000, 040214, and 041044 having approximately 8000 instances and 023031, 023034, and 040223 having around 4000. It is true that 040223 is the worst performer of the Brisbane group. However, 023031 has rather more instances than either 023034 or 040223, and consistently performs the most poorly. Therefore, we may hypothesise that dataset magnitude may be a factor on performance levels, but is less of a driver than data quality and other factors.

Significant intensity information is contained within daily climate datasets, above that supplied by daily rainfall measurements. With this information, more than half of the
4.7 Further Work

In proceeding with this work, the most obvious course would be to integrate the intensity information extracted from one or more climate records into a daily rainfall-runoff model. This would provide a strong indication of the utility of the method, and, as noted in Section 4.1, very likely result in an improved rainfall-runoff model.

It has also been noted that the comparison of multiple confusion matrices is laborious. Development of alternate methods of performance evaluation or presentation of the information contained in confusion matrices in a more elegant fashion would be of great use in this project and elsewhere. There are a number of possible avenues of inquiry in this direction, for example combining type I and type II errors into a single numerical measure, or presenting confusion matrices graphically.
CHAPTER 4. RAINFALL INTENSITY
Chapter 5

Modelling Water Quality in Jugiong, Muttama, and Tarcutta Catchments

In this chapter we consider the problem of modelling suspended sediment concentration as a function of streamflow magnitude. Suspended sediment modelling is important in the wider context of water quality modelling, and also for monitoring of soil erosion. Like the gully erosion example set out in Chapter 3, the application of classical techniques to the dataset in question met with limited success, so visualisation and data mining were used as further investigative tools. Where the difficulties in modelling density of gully erosion stemmed from the choice of density metric and the interdependencies within feature variables, in this case the limiting factor on model performance is sparsity of data.

The problem has two parts: the first, to discover a relationship between point measurements of flow and suspended sediment in three streams, and second to quantify the likenesses and differences in the three systems. A related issue is the conversion of concentration data into total loads, but in this chapter we shall focus on the relationship between suspended sediment concentration and flow magnitudes. Note that in this chapter we aim to produce numerical estimates of the target variable, rather than a discretised class as we did in Chapters 3 and 4.

Paucity of data imposes restrictions on the complexity of the models we can build. An efficient model design is one that uses all the available data, without performing unnecessary computational work or generating results of precision greater than that which can be supported by the data. Where data is scarce, these concerns are of great importance. The amount of data is limited, so it is clearly desirable to extract all possible information from what we do have, but we must also take care that the uncertainty that will arise
in data-poor modelling problems is properly accounted for. Therefore, we will keep all models generated in this chapter as simple as possible, adding complexity only when it is demonstrably productive to do so.

Model validation is also a concern where data is scarce, because little can be spared to form an independent testing dataset. In this chapter, the data sparsity problem is so serious that even ten-fold cross-validation (see Section 2.4.9) removes from analysis data that is needed for model building. For this reason, most of the performance measures listed below are taken from the fit of the training set to the models generated by that set. They are essentially measures of performance at the calibration stage, rather than validated quantities. Rather than building models for later application, the analysis detailed here seeks to investigate the nature of the streamflow - suspended sediment concentration system in these three catchments.

We begin as always by considering and discussing the data, in Section 5.1. The problem of noise within our limited dataset will prove to be an important factor, so it is considered in some detail in Section 5.2. Then, we proceed to basic visualisation and application of simple classical models in Section 5.3.

More focussed visualisation is performed in Section 5.4, where we use the qualitative insight gained by time series plots, scatter plots, and linear regression to consider more precisely the action of streamflow on suspended sediment concentration. The results of the Section 5.3 analysis can also be used as a benchmark against the knowledge generated by more complex or novel models may be evaluated (see Section 2.4.7). Recall that both visualisation and appropriate reporting of model results were among the principles set out in Chapter 1 and applied in Chapters 3 and 6.

In the later part of Section 5.4 (Section 5.4.1), additional variables are created in an attempt to better characterise the behaviour of streamflow. As noted above, where data is scarce it is doubly important to extract the maximum level of information, and in this case the pointwise time series representation of streamflow used in Section 5.3 was not capturing behaviour that visualisation suggested was important. The new variables describe the characteristics of the flow series around the points in time where suspended sediment concentration measurements were taken. Seasonal trends were also noted in the visualisation stages, so Season and Month variables were created, the former taking Summer, Autumn, Winter, and Spring values derived from the date markers associated with the data, and the latter being the month as a categorical variable.

Figures were then created comparing the nature of flow/suspended sediment concen-
5.1 DATA

...tration data pairs for different values of the new flow Shape and Season variables, and differences identified. The fact that the dataset can be usefully stratified according to these variables led to the adoption of a model tree approach (previously mentioned in Section 2.4.8). These models are similar in nature to the decision trees used in Chapters 3 and 4, except that instead of allocating a single value at each leaf, the model tree derives a real-number value from a sub-model. The result is a tree which finds and uses the most appropriate sub-model for a given instance. The sub-models we will use are linear regression models, in keeping with the theme of simplicity.

Section 5.5, regression model trees are discussed and constructed. The information gained from this approach is presented there, and compared with benchmarks obtained using regression on random partitions of the data. Alternative definition of the season variable is explored in Section 5.5.3.

As an alternative to model trees, a regression method that allows use of categorical variables was applied in Section 5.6. Several combinations of input variables were tested, so that we may identify those variables with high information content, and ultimately build the most efficient model. The subsequent section contains concluding remarks on this analysis and the other models applied. While the effectiveness of model building is limited by poor and scarce data in this case, we were able to discover useful knowledge regarding the flow - suspended sediment concentration system in these three catchments.

5.1 Data

Three catchments are considered: Jugiong; Muttama; and Tarcutta. All of these are located in inland South-Eastern New South Wales, Australia. While geographically close and similar in climate, the three represent a spectrum of catchment characteristics. Muttama catchment exhibits very little baseflow, with the flow falling to zero or near zero in the driest part of the year, while there is some baseflow in Jugiong, where very small flows do not occur in most years. The Tarcutta record demonstrates considerable baseflow. The three all have areas of the order 1000 of square kilometres, but response times vary due to the action of other physical catchment characteristics.

Water quality in the form of suspended sediment concentration was measured at semi-regular intervals for almost ten years from January 1994 to December 2003 for each of the three catchments. The number of points in each dataset were 103 for Jugiong, 102 for Muttama, and 108 for Tarcutta. For each catchment, there were a further few dozen...
measurements taken every 15 minutes or so for a handful of short periods. These event records brought the total number of points in each dataset to 184, 148, and 169 for Jugiong, Muttama and Tarcutta. Figures 5.1 to 5.3 are time series plots of water quality as suspended sediment concentration (hence referred to as $s_c$ or $wq$) and flow for the three catchments. For ease of visualisation, the Jugiong flow series has been reduced in scale by a factor of ten. Units of flow used are 0.1*ML/day. Suspended sediment concentration is given in mg/L, and values are derived from filtering and weighing sediment from water samples.

The $s_c$ measurements were most often about one month apart, or roughly fortnightly in the last fifth of the record. In several places, there are gaps of several months in the series. While the times of measurement were not the same for all catchments, they were typically similar. The sediment measurements were not taken at a consistent time of day, but streamflow was in all cases sampled close to (usually within an hour of) the appropriate time. The time step for all plotted streamflow throughout this chapter is a day.

In addition to the unevenly spaced water quality and flow data, daily streamflow time series exist for all catchments. These records, obtained from [110], are quite complete, with only one significant missing period in the Tarcutta record. The time of measurement here was consistently midnight as is conventional for Australian flow records. Comparison of the irregular flow data with the closest points (before and after) from the daily flow series
Figure 5.2: Muttama Water Quality and Flow Time Series

Figure 5.3: Tarcutta Water Quality and Flow Time Series
usually, but not always, yielded good correspondence. Quality of the flow data is discussed at length in the next section.

5.2 A Brief Discussion of Noise in Flow and Water Quality Data

For a few short periods in all three catchments, two different flow series are available. The first is the daily flow data obtained from the Pineena database ([110]) and the other higher resolution measurements covering a handful of disconnected events. The granularity of the second series ranges from twelve to six hours. The presence of a second, higher frequency flow record provides us with a basis for the estimation of uncertainty in flow data, even though the records are unlikely to be independently obtained. We may at least investigate how smooth the hydrograph is on a sub-daily scale, and thus infer how well a single daily value represents the behaviour of Jugiong, Muttama, and Tarcutta streamflow.

Where sub-daily flow data was available for comparison, the daily values were interpolated using a simple linear technique, between the two points on either side of the target, to produce an estimate of the flow at the sub-daily time of measurement. While the interpolation obviously introduces additional error, the technique has previously proved capable of providing useful estimates of daily rainfall totals shifted from 9am – 9am to midnight – midnight in Southern Queensland with minimal disruption. See [142] pp 58 – 70 for details.

It should also be noted that most flow measurements are taken at daily intervals, and if the model or models we seek are to be applied to situations where water quality data does not exist, it is likely that flow levels will be measured at midnight only. Therefore if we wish to obtain estimates at other times some adjustment will be necessary.

The percentage differences between the variable-time point measurements and fixed-time daily measurements are summarised in Table 5.1 below. The differences summarised below were taken using fixed-time daily point measurements as the baseline. The generating data for these values included between 48 (Muttama) and 70 (Jugiong) points for each catchment.

It is informative to note that where the variable-time measurements were taken close to midnight, the fixed sampling point, they were not necessarily closer to the corresponding estimated value than those sampled in the middle of the day. In fact, there are many cases where the sampling points were very close in time (less than 90 minutes) and the
reported flow magnitudes sometimes differed by up to or even greater than ± 30% in Jugiong, or ± 10% in Muttama and Tarcutta. Clearly, this demonstrates that the two measurement schemes returned significantly different values, especially in Jugiong. The average difference of -11.8% also represents a significant bias in one or both Jugiong series.

This brief exercise revealed significant issues surrounding the accuracy of flow records in the catchments under study. It serves to caution us against placing undue faith in the quality of such data in future sections and chapters.

Precision of $s_c$ is not quantitatively known, but [135] estimates that it may be as high as 30 or 40%. The amount of sediment suspended in a river is not constant across the stream, as flow velocity varies significantly within a channel cross-section, and local disturbances such as stock watering may muddy waters heavily for short periods of time. Clearly, uncertainty of this magnitude will have a significant impact on the quality of results it is possible to obtain from this data.

5.3 A Brief Note on Simple Models

Figures 5.4 to 5.6 are scatter plots of flow and $s_c$, on a log10-log10 scale for the sake of clarity. Clearly, the spread of water quality values for a single flow magnitude is very high. Some structure may exist, but it is very clear that a simple model, linear or otherwise, taking pointwise flow to $s_c$ will not produce a useful model in any case. A function $G$ such that $s_c = G(\text{Flow})$ would be the simplest model we would consider, and therefore the first to be applied, but as it is very unlikely that useful information would result from fitting a function of this form, we shall move on to a more sophisticated case directly. However, results from linear least squares fitting of this simple form will be used as benchmarks in later sections.

In Figure 5.6, two or possibly three points exist towards the upper left corner which may be classifiable as outliers. In particular the isolated point with very low flow and high suspended sediment concentration is intuitively unlikely. However this could be the

<table>
<thead>
<tr>
<th></th>
<th>Jugiong</th>
<th>Muttama</th>
<th>Tarcutta</th>
</tr>
</thead>
<tbody>
<tr>
<td>average</td>
<td>-11.8</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>max</td>
<td>42.3</td>
<td>13.5</td>
<td>26.7</td>
</tr>
<tr>
<td>min</td>
<td>-28.9</td>
<td>-53.8</td>
<td>-45.0</td>
</tr>
</tbody>
</table>

Table 5.1: Percentage Differences in Flow Values
result of vastly increased sediment concentration caused by stock watering or roadworks, and as the resolution of the water quality record is so low that the series contains no trace of persistence from one point to the next, we are unable to seek corroboration from neighbouring values. As we cannot say for certain that these points are erroneous, that shall remain in the analysis, but will also remain as a point for discussion.

5.4 Data Mining Approach and Visualisation

Beginning our approach to the problem in the usual way with visualisation, we arrive at much the same conclusions as [135]. First, the peaks in flow and sediment concentration occur roughly synchronised in time for all three catchments. In general, suspended sediment levels are much lower in Jugiong than Muttama or Tarcutta. The spread of sediment concentration values is by far greatest in the Muttama data. Figures 5.4 to 5.6 demonstrate that a relationship between flow and sediment concentration exists in that suspended sediment concentration seems to increase with flow, but the relationship is not sufficiently tight for quantification of this relationship to be useful as a model. In the following subsection, a deeper visual investigation is detailed.

Figures 5.1 to 5.3 show quite smooth progression in water quality from measurement to measurement. Due to the length of the interval between sampled points, this is unlikely to
5.4. DATA MINING APPROACH AND VISUALISATION

Figure 5.5: Muttama Flow vs $s_c$, log10 Scale

Figure 5.6: Tarcutta Flow vs $s_c$, log10 Scale
be the effect of physical persistence, but could be a result of seasonality of flow. Therefore it is likely that we shall have to deal with season in some way. For this reason, a month variable similar to that used the day-of-year Chapter 4 was created, as well as a season attribute taking values Summer, Autumn, Spring, and Winter as defined by the calendar.

The relationships extractable from the water quality/flow datasets by way of a function of the form $s_c = G(Flow)$ are not likely to be fruitful (see Section 5.3), so it is likely that there are processes at work that are not captured by simple point data. A more detailed look at a small section of the daily flow record with a few suspended sediment and flow point measurements gives some insight into the likely factors. Figures 5.7 to 5.9 show small sections from each of the three datasets.

The point measurements in Figures 5.7 to 5.9 are taken at slightly different levels of
flow, but the antecedent flow conditions are quite different in terms of the shape of the flow series around the sample point. In the work detailed in Section 5.3, it was assumed that suspended sediment concentrations $s_c$ could be expressed as a function of the flow $Q$ at that point $s_c = F(Q)$. From physical considerations supported and suggested by Figures 5.7 to 5.9, we may postulate that $s_c = F(Q, \text{shape})$ where \textit{shape} is a variable more completely representing the recent past, present and immediate future of the stream activity. For example, it is likely that the suspended sediment concentration of a sharp peak with a maximum at a flow of 100 ML/day will be greater than that recorded in the middle of a week of 100 ML/day flows. It is generally accepted that sediment concentrations on the rising and falling sides of individual peaks are significantly different ([28]).

Therefore, we introduce a variable describing the local shape of the streamflow profile. In this way, we hope to encode some of the information that a pointwise consideration of flow does not contain. For each point measurement, a shape was allocated by examination of the three preceding and one following value as well as the point flow measurement. The \textit{Shape} variable takes the following values, which were allocated manually and subjectively.

A flat or near-flat  
B decline  
C rise  
D peak  
E trough
Figure 5.10: Jugiong $\log_{10}(flow)$ vs $\log_{10}(s_c)$ coloured by shape

Figure 5.11: Muttama $\log_{10}(flow)$ vs $\log_{10}(s_c)$ coloured by shape

Figure 5.12: Tarcutta $\log_{10}(flow)$ vs $\log_{10}(s_c)$ coloured by shape
Figures 5.10 to 5.12 show the relationship between flow and suspended sediment concentration in log-log format, with each point coloured according to the basic shape.

No dramatic pattern emerges. Shape A (flat) tend to occur at lower flow values, which is not unexpected from physical considerations, but suspended sediment concentration remains comparable to values in other regions. Tarcutta data shows a tendency for $s_c$ to be greater when the flow profile is falling than when it is rising. This trend can also be observed to some degree in the Muttama data, although data points with a declining shape are more scattered there. In the Jugorng data, there is only the barest hint of this behaviour.

The information regarding magnitude of slope is not encoded into Figures 5.10 to 5.12—a small, gentle drop is indistinguishable from a large, sharp decline. Before launching into a discussion of slope, the exact definitions of each shape should be discussed. The classification of flow regions into shapes was performed manually to obtain the plots used above, by visual examination of the flow series. This method is clearly inadequate to deal with larger datasets, and quite time-consuming, so an automated shape classification method was built. The details are described in Section 5.4.1 below. The following figures were generated with the automatically extracted shape series.

Figures 5.13 to 5.15 are three dimensional plots of flow and local gradient vs suspended sediment concentration. Each point on the plot is coloured by season and has is marked by one of five devices that correspond to local shape. The plots were obtained with Matlab ([139]), and all were examined by rotating the frame of reference, but for the sake of comparison the frame of reference is kept constant in the figures below.

For reference, the breakdown of seasons and shapes by colour and symbol is given in Table 5.2.

These figures demonstrate that some structure is present. In Figure 5.13, there appears to be some separation between Autumn and Spring instances, and Autumn behaviour appears distinct in Figure 5.15. System characteristics are not the same for all catchments.
Figure 5.13: Jugiong Flow vs Local Gradient of Flow vs $s_c$

Figure 5.14: Muttama Flow vs Local Gradient of Flow vs $s_c$
Two-dimensional representations of three-dimensional plots are not in this very useful, although visualisation with rotatable versions of these plots yields more information. We may say that shape and season have some effect on the position of points in this space, but the order we see is by no means simple or easily quantifiable. Nor does it appear that we have a complete picture of the drivers behind suspended sediment concentration yet.

### 5.4.1 Automated Shape Extraction

The manually generated series of shapes formed a basis for comparison with the output of the automatic shape extraction routine, so that a fair degree of correspondence between the two could be guaranteed. For the shape series of all three catchments, this was achieved to a satisfactory degree— the output series differed from the manually generated series at very few points. However, we would ideally like an automated, repeatable, and less subjective routine for creating shape variables.

The shape of a region of the daily flow time series is dependent on the time scale (window size) considered. To account for this, two different shape lengths were used. Shapes in the first group were obtained only by consideration of the two closest points in time with the sampled point, effectively the midnight before and the midnight after the water quality
sample was taken. The second set of shapes were longer, calculated with points two days before and two days after the sampling day.

To be classified as 'flat', both forward and backward gradients were required to be small, but not precisely zero, in magnitude. After tuning by trial and error, the thresholds settled upon were 5 ML or 10% of the flow at the sampling point, whichever was larger. Unfortunately, at very low flows the 5 ML threshold may be too large, but almost all shapes at very low flows are flat (see Figures 5.13 to 5.15), so this problem is somewhat mitigated. These thresholds and the local shape over a single day gave correspondence with the manual shape series, even in the low-baseflow Muttama dataset. The wider window shapes were more inclined to be 'flat', but at this scale noise becomes an issue in the calculation of gradient, because the slope is rarely monotonic at the two day scale.

Where one of the forward and backward gradients was small compared to the other, it was disregarded in the the calculation of shape. For example, if the sampled point lay between a sharp drop on the left and a very slight rise on the right, the shape would be defined as a drop. The same 5 ML/10% thresholds were here to eliminate relatively minor variations to one side.

Comparing the two shape series, it was found that the local gradients (those obtained with a two day window- one day on either side of the point in question) yielded a more useful shape classification than those obtained with a wider range (two days on either side). Of most choices of parameter, the local series corresponded better to the manually generated series. Although some useful information is probably lost, the wide window method was abandoned and only the local shape variable used in further analysis. All of the study catchments are relatively small in area and have fast response times- a wider window may be more appropriate for larger catchments.

To account for magnitude, another variable (Mag) must be introduced. Where the shape is rising, or falling the definition of magnitude is naturally the difference between the end points of the shape window, and can be assumed zero for flat shapes, but where the shape is either a peak or trough, the difference between endpoints is not informative. Rather, the average magnitude of the forward and backward differences from the water quality measurement point was taken. This formulation would be problematic in the situation where one of the left or right gradients is vastly larger than the other, but the 10% threshold discussed above comes into play in the most extreme of these cases.

There is some variability in the time of day the water quality measurement was taken, and this will lead to differing treatment of forward and backward gradients. However, as
most measurements were taken manually, it is safe to assume that few points lie close to
the midnight boundary between days. A quick inspection of the times reveals that almost
all lie between 0.35 and 0.70 on the decimalised time scale, or in other words about 8:30am
and 5:00pm. Thus we conclude that the ill-effects of this phenomenon are not likely to be
significant.

The issue of the position of the water quality sample arises again in that the sample
point may be at the end of a shape, in the middle, or at the beginning. The distance to
peak variable $p_{\text{dist}}$ was introduced to counter this variation. Very simply, the preceding
peak is defined as the maximum flow in the last five days. $p_{\text{dist}}$ is the number of days
between the preceding peak and the $s_c$ measurement plus one, or $p_{\text{dist}} = 0$ if the water
quality measurement point itself is the maximum.

Some other quantities naturally appeared out of the peak distance calculations. These
were $p_{\text{max}}$, the height of the greatest flow peak within the five day interval, $p_{\text{fall}}$ the
difference between this and the measured flow value, and $p_{\text{grad}} = \frac{p_{\text{fall}}}{p_{\text{dist}}}$ when $p_{\text{dist}} \neq 0$. These variables are adopted in Section 5.6.

One aspect of the flow system not captured by the set of descriptive variables as it
stands is magnitude of event. No distinction has been made between a peak of 10 and
100 ML, for example. Magnitude information will be included in Section 5.6, but for the
purposes of stratification (splitting the dataset), it is undesirable to introduce any further
distinctions between shapes, because the size of the dataset is small and the more pieces
it is divided into, the less data is contained in each partition.

5.5 Model Structure

The figures in Section 5.4 imply that the relationship between water quality and flow
magnitude varies depending on the local shape of the hydrograph and the time of year,
although these two factors are by no means independent. Therefore the model at this point
should be some function $s_c = G(\text{Flow, Shape, Season})$. Taking the most obvious option,
the datasets could be divided into groups (or 'cohorts') according to season and local shape
and simple functions could be fitted to each cohort. Note that this process is equivalent
to the introduction of new variables describing flow profile and season. While the model
extracted using the additional information thus introduced would still be very imprecise,
it is likely to be better than the simple least squares fit discussed in Section 5.3.

However, partitioning the dataset into many smaller pieces has its own disadvantages.
Chief among these is the increased uncertainty that comes when fitting any curve to less data. Assigning a new cohort for each unique combination of season and local shape, twenty cohorts are created. In this way a tree of models will be created. Automated model tree builders do exist, such as the M5' (pronounced em-five-prime) algorithm mentioned in Section 2.4.8, but due to small number of variables involved in this problem and the physically intuitive choice of season and shape to create cohorts, the model trees built below are generated with simpler methods.

The datasets for each catchment consist of about 180 points each, so each cohort will be populated by an average of only around nine data points. The effect of the data partitioning is similar in nature to the problem faced by decision trees, where each split successively shrinks the data available for the algorithm to work with. The result may be an unrealistic model that appears much better than it is - in validation, performance will likely plummet if this is the case.

Ironically, it is because of data scarcity that a section of the water quality/flow data is not kept back and used for validation. Instead, other methods will be used to account for the effects mentioned above as far as is possible. See the next section for details.

Perhaps the best strategy for controlling overfitting and complexity issues is to keep the model as simple as possible. To this end, it may be appropriate to aggregate cohorts with similar populations into one. Initially however, the full twenty-cohort scheme will be examined along with simple stratification by season and local shape alone.

### 5.5.1 Preliminary Model Trees

Table 5.3 gives the absolute total errors generated by the various model schemes. 'Universal' denotes use of a single universal cohort represented by one linear least squares (LLS) fit through all the data; 'season only' denotes a tree with splits on *Season* only, and similarly 'shape only' trees are built by splitting on *Shape* only to the tree with splits on shape and season. 'Season and shape' trees allow splitting on both variables.

<table>
<thead>
<tr>
<th></th>
<th>Jugiong</th>
<th>Muttama</th>
<th>Tarcutta</th>
</tr>
</thead>
<tbody>
<tr>
<td>universal</td>
<td>0.544</td>
<td>0.479</td>
<td>3.05</td>
</tr>
<tr>
<td>season only</td>
<td>0.495</td>
<td>0.288</td>
<td>2.99</td>
</tr>
<tr>
<td>shape only</td>
<td>0.498</td>
<td>0.351</td>
<td>3.01</td>
</tr>
<tr>
<td>season and shape</td>
<td>0.440</td>
<td>0.240</td>
<td>2.89</td>
</tr>
</tbody>
</table>

Table 5.3: Preliminary Absolute Errors for Different Tree Schemes, times $10^4$
5.5. **MODEL STRUCTURE**

Here and in the following sections, the term 'absolute error' refers to the sum of the absolute values of the discrepancies between observed water quality value and the corresponding value from the modelled series, \( \delta_i = |s_{c_i, \text{obs}} - s_{c_i, \text{mod}}| \) and absolute error = \( \sum \delta_i \).

From Table 5.3 it seems that the three model tree methods have a significant advantage over universal LLS fitting in the Muttama catchment, a smaller advantage in Jugiong, and very little to commend it in Tarcutta. Errors for Tarcutta given in Table 5.3 are an order of magnitude larger than those for the other catchments, but the number of data points is similar for all catchments. Recall that three possible outliers where noted in Tarcutta data in Section 5.3. It is possible that these points are responsible for generating a large part of the errors reported for Tarcutta, and we shall investigate this idea later in this section.

Recall that we noted in Section 5.3 that any model taking the form \( s_c = G(\text{Flow}) \), as the universal LLS approach does, was unlikely to produce useful information. However, the error measurements for the universal and model tree schemes are not directly comparable. The model tree approach error totals are likely to be considerably less than those obtained from the universal scheme for the simple reason that the number of points generating each fitted line is much less, and hence the estimates are likely to be closer to the data points. The apparent rise in accuracy is paid for with a sacrifice of certainty, because each sub-model is based on far fewer data points than the universal one.

Uncertainty is unfortunately hard to quantify, but must be related to the quality of the input data, which in this case is not high. With larger datasets, cross-validation would certainly be performed, but in this case, where we have less than 200 data points for each catchment, the value in separating the data into cohorts is explored by randomising the contents of the cohorts and using this case as a benchmark. Keeping the number of points in each cohort constant, the allocation of points to cohorts was randomised, so that no partition has homogeneous season or shape. This allows us to identify the change in absolute error resulting from reducing the size of the dataset being modelled at each leaf. The new cohorts were modelled and evaluated using the same simple linear fitting process as the models reported above. The experiment was repeated five times, and the results are summarised in Table 5.4 below.

Comparison of Tables 5.4 and 5.3 is especially notable for the Muttama catchment. With this dataset the model tree method presents a considerable decrease in error over the baseline model, but in Jugiong and Tarcutta the shape-season model tree error is not significantly smaller than that obtained with the contents of each cohort randomised. In fact, it is slightly larger.
Table 5.4: Absolute Errors from Shape and Season Randomisation Experiment, times $10^4$

<table>
<thead>
<tr>
<th></th>
<th>Jugiong</th>
<th>Muttama</th>
<th>Tarcutta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 1</td>
<td>0.477</td>
<td>0.366</td>
<td>2.26</td>
</tr>
<tr>
<td>Run 2</td>
<td>0.414</td>
<td>0.331</td>
<td>2.41</td>
</tr>
<tr>
<td>Run 3</td>
<td>0.442</td>
<td>0.387</td>
<td>3.07</td>
</tr>
<tr>
<td>Run 4</td>
<td>0.455</td>
<td>0.359</td>
<td>2.60</td>
</tr>
<tr>
<td>Run 5</td>
<td>0.476</td>
<td>0.361</td>
<td>3.22</td>
</tr>
<tr>
<td>Mean</td>
<td>0.453</td>
<td>0.361</td>
<td>2.71</td>
</tr>
</tbody>
</table>

The small size of the three datasets means it is likely that extreme events are poorly represented. An examination of Figures 5.1 to 5.3 reveals that the initial Jugiong dataset in particular contains one point that is far greater in both flow and suspended sediment concentration than the general population. Figure 5.3 shows one point with flow in the usual range, and the corresponding $s_c$ is also very high. Because of the sparsity of information in these regions and the extreme nature of the data values, any model that does not take special account of outliers will either model extreme points very badly or be skewed by them. As we are using a least-squares approach, the possibility of outliers skewing the model is strong. Therefore, we shall briefly experiment with removing them.

We also noticed outliers in the Tarcutta dataset in Section 5.3. The extended datasets were filtered to remove the most obvious outliers from all three datasets. As a rule of thumb, any point whose distance (in either flow or water quality variables) from the nearest non-outlying point was greater than the width of the population was removed. This resulted in the erasure of three points from the Jugiong and Muttama datasets, and two of the three possible outliers from Tarcutta. Note that not all of these points are shown in Figures 5.1 to 5.3, because these figures show only the smaller initial dataset used for visualisation. The fitting process was then re-run, with results as shown in Table 5.5.

These results show much the same trend as those presented in Tables 5.3 and 5.4. Only in the case of Muttama and to a lesser degree Jugiong, does the model tree method appear useful. However, we shall see in the next section that the model tree method does provide us with useful information.

### 5.5.2 Model Trees, Error and Uncertainty

The models built and discussed in Sections 5.5.1 and below were designed to minimise the sum of the squares of the distance between predicted and observed water quality values
Table 5.5: Outlier-Removed Data, Absolute Errors, times 10^4

<table>
<thead>
<tr>
<th></th>
<th>Jugiong</th>
<th>Muttama</th>
<th>Tarcutta</th>
</tr>
</thead>
<tbody>
<tr>
<td>universal</td>
<td>0.411</td>
<td>0.378</td>
<td>1.95</td>
</tr>
<tr>
<td>season and shape</td>
<td>0.322</td>
<td>0.218</td>
<td>1.89</td>
</tr>
<tr>
<td>random run 1</td>
<td>0.323</td>
<td>0.314</td>
<td>1.91</td>
</tr>
<tr>
<td>random run 2</td>
<td>0.343</td>
<td>0.317</td>
<td>1.94</td>
</tr>
<tr>
<td>random run 3</td>
<td>0.327</td>
<td>0.302</td>
<td>1.99</td>
</tr>
<tr>
<td>random run 4</td>
<td>0.345</td>
<td>0.311</td>
<td>1.97</td>
</tr>
<tr>
<td>random run 5</td>
<td>0.330</td>
<td>0.305</td>
<td>1.95</td>
</tr>
<tr>
<td>random mean</td>
<td>0.337</td>
<td>0.310</td>
<td>1.95</td>
</tr>
</tbody>
</table>

and reported in terms of the total absolute error, but absolute errors (or squares of the same) are not the whole picture of error behaviour. We must also consider how percentage errors behave. For example, if the distance between predicted and observed values was equal to 10 mg/L for each point, then the smaller water quality values would have an enormous percentage error and the larger values a very acceptable tiny percentage error. On the other hand, a uniform percentage error would mean most of the absolute errors are contributed by larger water quality values.

By the same token, average errors in a region may be near zero even though the spread of values on either side of the predicted line is high. Of course the opposite also applies in that the average magnitude of error may be small for each point, but if all errors have the same sign, they may also appear larger when considered in total. For this reason, use absolute error values in our analysis.

Care should be taken with the word 'error' in all contexts, so we will state explicitly what we mean by it here. When we say the error or errors of some model are such and such, we are not explicitly making a comment on predictive capacity or even correctness of model. As written here, error is merely the distance between observed s_e values and the value output from the model, or the aggregate of these distances. It is a measure of how closely the model reproduces the input data, not an estimated predictive accuracy of the model.

As demonstrated by Table 5.3, the gains in goodness of fit made by model trees are small, but significant for Muttama and perhaps also for Jugiong catchments. Even in the Tarcutta catchment, accuracy can be lifted by a tree splitting on both Season and Shape. However, the randomisation experiment reported in Table 5.4 implies that some of this gain, particularly for Tarcutta, is due to the reduced number of points fitted by each sub-
model rather than the capture of useful information. For at least Muttama and Jugiong, it can be stated that some system information has been successfully captured and exploited, but we are still far from a in-depth understanding of the behaviour of $s_c$.

To better codify our system knowledge, we shall examine the accuracy of individual sub-models. In this way we may discover regions of Shape and Season where the relationship between flow magnitude and water quality is clearer (and can be expressed with more certainty). Of particular interest is the case where flow profile is flat, because this implies a stable state. If we cannot uniquely identify a water quality value for a given flat shaped flow magnitude, it is clear that other factors are at work that we have not captured with our new variables.

The breakdown of absolute error accumulation ($\delta$ is the sum of all $\delta_i$ for all points with common season and shape) in each cohort was examined, along with number of points in that collection and the two parameters obtained from least squares regression for the cohort. While the tables of results are not reproduced here for the reason that they are extremely large and unwieldy, several conclusions can be drawn from detailed examination of this information. First among these points is the trend for flat cohorts to be more neatly fitted than their counterparts of other flow profile.

Aside from the underpopulated cohorts where the error estimate is of course unreliable, it is quite clear that the simple linear model fits quite well within some season-shape groupings, and very poorly in others. For example in the Jugiong dataset, the total error for the 17 flat autumn points was 34.5, or just over 2 mg/L per point, where the 18 rising winter points contribute 189.0 mg/L to the total error, or 10.5 mg/L per point. Thus we can state that in some partitions good results can be obtained. Also notable is the spring trough cohort, where the fit is particularly bad and the 19 data points highly scattered, contribute a total of 2476.2 units to the overall error, or more than 130 mg/L per point on average. Detailed visual examination and statistical analysis of each individual cohort revealed that in a few cases much of the error in a given submodel was due to one or two local outliers, but in most poorly performing cohorts a simple relationship between flow magnitude and $s_c$ just did not exist.

General trends across the three catchments can be made regarding linearity in particular cohorts. With the exception of Tarcutta and Muttama summer data, flat shapes give rise to reasonable errors, between about 2 and 10 mg/L. These values seem reasonable for a useful model. Some bias may be introduced from the fact that a disproportionate number of flat shaped events are small in both flow and water quality, as in many cases they represent
the ground state of the system, undisturbed by rainfall and subsequent sediment arrival. Because flat flow profiles tend to belong to low flow magnitudes, uncertainty is greater in higher magnitude regions. However, all of these cohorts are relatively populous, and the mean errors here are in keeping with the uncertainty in suspended sediment measurements noted by [134].

However, it is also true that even for flat profiles of a given season, significant scatter occurs in a number of regions. As noted previously, this may imply that there are significant factors at work that are not captured by the flow and flow shape variables- if a flat-profile flow magnitude of 10 ML/day corresponds to a wide range of water quality values, either the uncertainty in $s_c$ is very high, or something else is influencing suspended sediment concentration.

Rising and falling data seems moderately non-linear, except in the case of Tarcutta summer and autumn, where the fit is unusually poor. Trough and peak data is sparse, so we will refrain from making generalisations in this area.

Looking at seasonal trends, summers (again with the exception of Tarcutta) are variably modelled, with a range of error rates. Spring events are well modelled where the flow shape is not falling, with average errors of around 10 mg/L per point. Winters in Tarcutta and Muttama fare a little better then 10 units per point for rising, flat, and falling shapes, as do autumn events.

### 5.5.3 Season Definition Experiment

A number of other experiments were performed using these datasets. The most interesting of these centred around the definition of seasons. Instead of the simple Summer, Autumn, Winter, Spring division, qualitatively different periods were isolated manually. This was done very simply by visual means, by plotting the series against day of year from one to 365, so that every year was overlaid on the others. Breakpoints were then inserted where them seemed to best divide the events of the year into characteristic groups.

The new boundaries are listed in Table 5.7 below in terms of day or year. Table 5.6 shows the boundaries of calendar seasons for reference purposes. In both tables, boundaries are given for non-leap years.

Clearly, for all three catchments the manually defined periods correspond roughly to Summer, Autumn, and Winter-Spring. Note that although three periods were defined in each case, no apriori decision regarding number of periods was made. Table 5.8 below summarises the total absolute errors obtained from the model trees created with seasons
defined as in Table 5.7, and mean total absolute error of five randomised experiments and
the same statistics for the simple universal regression model.

Compared to the preliminary model tree results listed in Section 5.3, these results show
a significant drop in accuracy of fit for the Muttama catchment, and some small loss for
Tarcutta. The Jugiong statistic is identical to three significant figures. However, if we
also compare the above with Table 5.4 it is evident that the mean error levels for the
randomisation experiment have risen, except for Tarcutta data. This is due to the reduced
number of cohorts produced by the reduced number of seasons. For Jugiong, Muttama,
and Tarcutta, the number of non-empty cohorts with manually defined seasons were 14,
13, and 14 respectively. Universal figures are of course the same.

The results shown in Table 5.8 imply that the manually-defined seasons may produce
better or worse models than calendar seasons, but in no case was a significant gain made,
so this approach was abandoned. In the remainder of the experiments detailed in this
chapter, the calendar seasons are used.

5.6 Regression with Additional Features

Although the overall appropriateness of fit did not in general greatly improve between
the baseline universal least squares model and the cohort-separated model tree, the latter

<table>
<thead>
<tr>
<th></th>
<th>Jugiong</th>
<th>Muttama</th>
<th>Tarcutta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model tree</td>
<td>0.440</td>
<td>0.306</td>
<td>2.43</td>
</tr>
<tr>
<td>Universal</td>
<td>0.544</td>
<td>0.479</td>
<td>3.04</td>
</tr>
<tr>
<td>Mean random</td>
<td>0.469</td>
<td>0.382</td>
<td>2.47</td>
</tr>
</tbody>
</table>

Table 5.8: Absolute Errors from Season Definition Experiment, times 10^4
5.6. *REGRESSION WITH ADDITIONAL FEATURES*

approach has the advantage of providing additional information regarding likely uncertainties. The presence of such strong structure in the errors implies that neither model was accounting for system behaviour in a satisfactory manner.

With this in mind, a second modelling approach was adopted, building on the lessons of the first two. First, the errors generated by the baseline LLS model were run through a linear regression model which was passed the descriptive variables as inputs. Ideally, this would expose which of these variables could be used to refine the simple model and obtain a better one. The obvious question at this point is why we bother modelling errors at all, if we are already undertaking a regression exercise with some form of variable selection, and in fact the two activities are one and the same. However, due to the massively non-unique nature of the possible optimal models, the two approaches were applied and different results obtained.

An advantage of adopting a regression approach is that more variables can be introduced with relative ease. While we could have used these variables in the model tree scheme, the partitioning system we used does not leave enough data in each node to support additional complexity. Regression, where the data is treated as a block, does not suffer from data scarcity problems so much. Because each line of best fit is drawn taking all points into account, variation caused by noise will be averaged out, but where many lines are being fitted, each to only a small number of points, the effect of noise can be more detrimental.

The new variables are those discussed in Section 5.4.1 $P_{max}$, $P_{dist}$, $P_{max}$, $P_{fall}$, $P_{grad}$, and $Mag$. In addition, the left and right gradients $Rg$ and $Lg$ were introduced into the dataset.

These models were obtained with the linear regression package in the Weka suite ([112]), and although two attribute selection algorithms (see Section 2.4.5) are available for use with this package, a number of combinations of variables were manually selected for trial. In all tested cases these two feature selection algorithms, an implementation of the technique used in the M5' algorithm (see Section 2.4.8) and a greedy method, produced the same result in negligible time, so the default M5' option was selected.

The combinations of variables were as follows:

- **All** All available attributes
- **P** As above, with $P_{max}$, $P_{dist}$, $P_{max}$, $P_{fall}$, and $P_{grad}$ removed.
- **M** Only flow, shape, and season included
- **Q** Flow only
The first collection involved all the information we could gather about the system, while the second neglected to include some of the variables whose relevance has not yet been demonstrated. The third combination, -M, consists of the basic set of variables used in building model trees, and then we have the set Q, which contains flow alone. This last experiment is included as a benchmark.

Water quality was used as the target for regression with each of the four attribute collections. The results for the Q collection should of course be much the same as the baseline universal model used in Section 5.5.1, and a quick check reveals that the models are similar, and produce a similar level of error.

We noted in Section 5.5.2 that the spread of errors is not even throughout the dataset. In some regions in feature space the suspended sediment concentration values appear to be linear with flow, and in other places are scattered apparently randomly. Unfortunately, the thinness of the dataset in terms of number of points makes it difficult to make our observation quantitative, but we will try to do this in a simple way by modelling the errors output from the model tree and universal least squares model by regression. In this way, the structure present in the errors should be examinable, even if our conclusions must necessarily be uncertain when drawn from a small pool of observational data. We could of course have visualised these errors, but the regression approach provides a more compact numerical representation.

When separate regression models are built for each combination of input attributes with percentage, absolute, and raw error, the number of model runs becomes quite significant. We have four input feature groupings and seven target variables including water quality and errors generated from the universal baseline model and the season and shape model tree, making 28 runs in all for Jugiong, Muttama, and Tarcutta, or a total of 84 modelling exercises. Happily, each run takes only a few seconds to complete. This is of course the upside of working with small datasets, but the output parameters for all variables and all runs makes a formidable table for each catchment, and for this reason these tables are not reproduced here.

Listed below are the water quality models generated with all available features as input.

\[
WQ_j = 0.354Q + 65.217(\text{Shape} = \text{Peak}) - 5.822p_{dist} \\
+ 22.911(\text{Season} = \text{Winter, Spring}) + 17.109
\] (5.1)
5.6. **REGRESSION WITH ADDITIONAL FEATURES**

\[ WQ_M = -0.1566 Lg - 0.1389 Rg + 23.1942 (\text{Shape} = \text{Rising, Falling, Trough}) \]
\[ + 68.2001 (\text{Shape} = \text{Peak}) + 0.1743 Mag + 0.0427 p_{\text{grad}} \]
\[ + 26.2064 (\text{Season} = \text{Autumn, Spring}) \]
\[ + 48.607 (\text{Season} = \text{Summer}) + 5.8106 \]  (5.2)

\[ WQ_T = 0.1973 Q - 0.1672 Lg + 0.1585 Mag - 67.2779 p_{\text{dist}} \]
\[ - 0.3662 p_{\text{grad}} + 228.4329 (\text{Season} = \text{Summer}) + 186.7 \]  (5.3)

Of course, none of these models perfectly represent the available data. There is a small but significant increase in accuracy for representation when additional variables are introduced, but we are not concerned with predictive capacity here except in a very limited sense. More interesting is the structure of the output models, and what this tells us about the internal dynamics of the system that relates catchment water quality to flow.

Results are not tabulated here, as they are extremely extensive, but a number of interesting trends appear in these models. Strong parameters for a particular variable in the model estimating \( WQ \) imply a dependence on that variable. If the parameters corresponding to that variable also show up in the error estimating models, this means that the baseline models are not capturing the full range of behaviour where that particular variable is acting. Thus, we obtain some idea of how the additional variables act in the system, as well as a more general view of what the universal baseline model misses.

As expected, regression cannot find any structure in overall error with just the season, shape, and \( Q \) (flow) variables. If there was structure identifiable in this case, it would imply a serious problem with the model tree builder. While linear regression with Boolean dependence on categorical variables allows for some differences between feature space regions, it is limited to adding a constant value- see the regression models above for examples. The model tree structure allows for a change in the gradient between \( Q \) and \( s_c \) between partitions. More trends noted from the regression modelling levels are given in the following subsection, along with a few accompanying remarks the action of different variables.
5.6.1 Results from Regression Experiments

Seasonal trends seem to differ between catchments. In Jugiong, Winter, Summer and Autumn tend to have less sediment in stream compared with flow, and almost uniformly, the error measures for Jugiong show a drop in accuracy in Spring where the ratio of suspended sediment concentration to flow is higher. Winter has the lowest ratio of sediment to flow in all catchments, but this may be due to the existence of larger flow levels at this time. Summer is perhaps the most predictable season. Conversely, the other two datasets show that the ratio between $s_c$ and $Q$ rises in Summer and to a lesser degree in Autumn. The $Season = Summer$ condition also implies a rise in most error measurements.

A negative dependence on $p_{dist}$ exists, and can be seen in Jugiong experiments with as a factor of about -5 and in Tarcutta as around -50. This fits well with the qualitative idea that a peak in streamflow is accompanied by a peak in suspended sediment concentration, which then both recede.

There is a distinct negative dependence on both $p_{dist}$ and $p_{grad}$ in Tarcutta data, which elaborates on the above hypothesis slightly in that sharpness of peak seems also to have a negative effect on the value of $s_c$. This seems curious, but the overly simplistic calculation of $p_{grad}$ may explain it. Recall that $p_{grad}$ is not a true estimate of the flow gradient, but simply the difference between the flow at the point in question and the peak value divided by the number of days between the two. When the local profile is rising, the $p_{grad}$ estimate is likely to be low. Thus the negative factor on $p_{grad}$ will contribute less. It is likely that the interaction between the two active peak variables here is also affected by local shape.

Muttama catchment in particular shows that points with a flat flow shape are lesser in both suspended sediment concentration and error, which is what we would expect. Peaks tend to have a greater $s_c$ value than other shapes. In Tarcutta the trends are somewhat less clear, but rising flow profiles add to the expected $s_c$ value where peak variables are disallowed from the regression (see models listed above). Otherwise, this aspect of the system is represented by a negative dependence on $p_{dist}$ and $p_{grad}$. The Jugiong dataset shows clearly that peak points have a greater $s_c$ value than would otherwise be expected, as well as an unsurprising reduced accuracy in the universal model.

The water quality estimators return a flow parameter of around 0.35 for Jugiong and 0.20-0.25 for Tarcutta. Interestingly, it is Tarcutta that has the greater flow, and is the only one of the three catchments with a significant baseflow component in that flow. In Muttama the dependence on flow is much less distinct, only showing up when all variables apart from shape, season, and $q$ are disallowed, but the action may be represented by a
positive dependence on $Mag$ instead. As we noted in the visualisation stage, Muttama has by far the greatest sediment loads in comparison to level of flow.

Other variables of interest include $Lg$ which tends (particularly in Muttama and Tarcutta) to have a negative relation with water quality. Less clear is a tendency towards a negative factor on $Rq$ in Muttama and to a lesser degree Tarcutta. This indicates that if the profile is rising, $s_c$ is lower. This finding is in contract to the observation of Section 5.4, where we noted that $s_c$ tends to be higher for rising profiles than for falling. A factor of about 0.20 of magnitude of event $Mag$ contributes to $WQ$ in Muttama. This parameter falls to about 0.16 in Tarcutta, and is not recognised at all in Jugiong. The variables $p_{fall}, p_{max}$ were rarely if ever identified as important. The only places either of these two showed up with a non-zero factor were in Muttama catchment, where the system seems most confused under this representation.

### 5.7 Conclusions

The various experiments performed on the Jugiong, Muttama, and Tarcutta suspended sediment concentration data yielded a number of interesting conclusions, if not a usable model taking point flow to $s_c$. First, we examined the data visually with time series and feature space plots and found that the relationship between point flow and water quality was not constant. Thus, the model in question evolved from $s_c = F(Q)$ to $s_c = F(Q, \text{shape})$ to $s_c = F(Q, \text{shape, season})$ to $s_c = F(Q, \text{shape, season, ...})$.

The model tree structure had to be abandoned due to scarcity of data when it became clear that more factors than just season and local flow shape were at work, in favour of the slightly more restrictive but much more data-efficient linear regression with with Boolean variables approach. Some gains in accuracy over the baseline model were achieved, although data scarcity prevented us from testing any of the models created with these paradigms with satisfactory rigour. We also now have some idea of where we can and cannot predict $s_c$ with reasonable accuracy, although these hypotheses are of course subject to uncertainty.

At the very least, significant insight into the nature of the system was achieved. We now have a bare outline of a conceptual suspended sediment concentration model that takes into account season (whether calendar or otherwise defined), local flow profile, and a few relevant antecedent flow conditions. We have also demonstrated the use of model trees and other techniques, for water quality problems. Differences between the seasonal behaviour of catchments were identified- if not in a precise fashion- and the relationship
between the general characteristics of the hydrograph and the form of the dependencies on more local flow profile variables has also begun to emerge.

What first appeared to be simply noise has revealed itself as complex structure based on much more than just point flow. This is the achievement of this project, and represents a significant leap in knowledge and sophistication from the position adopted at the beginning of this chapter.

In the next chapter we shall explore another, different measure of water quality for a location not far from Jugiong, Muttama, and Tarcutta. There, the water quality sampling interval is only 15 minutes, and the uncertainty very low. Ideas developed here are further explored, and the quality of result much greater.
Chapter 6

Modelling Electrical Conductivity in the Yarrangobilly Catchment

6.1 Introduction

Here we continue to investigate the relationship between flow and water quality using techniques from data mining. In Chapter 5, the small number of suspended sediment concentration data points available to us was a problem, as was the high uncertainty of each point. Verification of the model on an independent test period was not possible especially for the model tree method. Many of the cohorts were also very small or even empty, which increases the uncertainty of model outputs in these areas. We do know at which points the uncertainty is small and at which it is not small, which is an advantage of the model tree approach over the universal linear model. But in order to judge further the correctness of our approach, additional testing should be undertaken. To this end, further data was obtained from a catchment about 100 km from the three discussed in Chapter 5. Rather than suspended sediment concentration, another type of water quality measurement was available for use.

The relationship between streamflow $q$ and electrical conductivity $k$ is explored in this Chapter, using data from Hollin Cave Spring in New South Wales, Australia. A temporal rule extraction algorithm is used to identify frequent patterns in each time series. This is consistent with the knowledge extracted from the previous study on suspended sediment concentration, where the value of an event-based became apparent. Temporal rule extraction is possible in this case where it was not in Chapter 5, because the quality, resolution, and quantity of available data is in this case much higher for flow and water
quality series. It has the advantage of a physically interpretable, compact output, and fits the nature of the problem well.

The frequent patterns identified by the rule extraction algorithms are then refined using the concept of profile convexity (an advance over the flat/rising/falling concepts used in Chapter 5), and parametrised for compactness of representation, before examination of the coupling between flow and conductivity. Results show that two frequent peak patterns occur in flow and two troughs in electrical conductivity, and that the shapes of all these can be characterised with a single magnitude parameter. The coupling between events in the two series is investigated, and reveals that the depth of $k$ troughs depend heavily on the initial state of $k$, and more weakly on the magnitude of the flow peak.

Electrical conductivity $k$ provides an estimate of dissolved solids in-stream, as the relationship between salinity and electrical conductivity in water is well established. Throughout this chapter, numerical units of $k$ are $\mu$S/cm (micro-Siemens per centimetre), and are Litres/second for $q$. Compared to most other water quality indicators such as suspended sediment load, it is easy to measure. Where some contaminant concentrations rise after rainfall due to increased input, $k$ will fall as the salt load delivered by groundwater is diluted by a greater volume of streamflow or discharge $q$.

The data and study area are discussed in Section 6.2, along with some motivation on the problem of electrical conductivity modelling. New variables created from the flow and electrical conductivity series are detailed in Section 6.3, which runs parallel to Section 5.4 in the previous chapter. The idea developed there are applied here as well, to a richer dataset where further analysis is more likely to produce useful results. The most complex objects defined in the Jugiong, Muttama, and Tarcutta flow series were peaks and troughs formed by an up-down or down-up pattern, but here we extend these concepts in include more complex peaks and troughs.

Rather than manually define a set of shape categories, we shall use a modified version of the Apriori rule extraction algorithm (see Section 2.5.2 or the discussion below) to produce a set of peak and trough shapes that occur frequently (a quantitative definition of frequency is contained in Section 6.5). The additional complexity introduced by the extended representation of shape is thus justified. The frequent shape (or motif) discovery algorithm is discussed in Section 6.4.

The extracted shapes are described in Section 6.5. Three frequently occurring peak shapes are identified from the flow series, and four trough patterns from the conductivity series- basic visualisation in Section 6.2 revealed that the typical response of electrical
conductivity to a streamflow peak is a sudden drop and slow recovery. In Section 6.6, the magnitude and internal variation of the frequent patterns is investigated, and it is found that while motifs of different magnitudes are not congruent, the magnitude of each component rise or fall can be roughly determined by the total magnitude.

The relationship between streamflow event and the response of in-stream electrical conductivity is discussed in Section 6.7, when the representation of both series are well developed. A relationship between corresponding flow and conductivity is established there. Unexpectedly, the best representation of this relationship is between magnitude of flow peak and minimum point in the conductivity event, with a far weaker dependence on initial state of electrical conductivity. Initial state of the flow series appears to have relatively little effect.

Certain anomalies in the timing of events also arise. In a few cases, the conductivity event appears to begin before the only nearby streamflow event. This, and other physical points of note, are discussed in Section 6.7 as well, and also in Section 6.8, which houses general concluding remarks.

6.2 Description of Available Data

Hollin Cave catchment studied here lies in the Snowy Mountains of South Eastern Australia, entirely within the Yarrangobilly Limestone. The catchment is small (less than 500 Ha) and very steep. More than 300 caves and numerous hot and warm springs have been identified in the area [140]. Hollin Cave Creek is connected to other streams underground in Eagle’s Nest Cave, one of the deepest cave complexes on the Australian mainland. Data were collected from this system in the hope of better understanding the karst hydrology regimes.

The Hollin Cave Spring dataset consists of 31341 paired data points for $q$ in Litres/second, and $k$ in micro-Siemens/cm at half-hour intervals with only a few breaks over a three year record. Almost all of these breaks are short, of the order of a few hours. The apparatus is discussed at length in [141]. Electrical conductivity and flow (discretised as discussed in Section 6.3) are plotted over time and then against one another in Figure 6.1 and Figure 6.2, which demonstrate that the behaviour of $k$ is not solely controlled by volume of flow. However, as a starting point we shall assume that it is and create and calibrate a linear model to use as a baseline for comparison with subsequent schemes. Linear regression yields $k = -0.41q + 230$. Using ten-fold cross-validation, we find that the average mean
absolute error in $k$ is 31 micro-Siemens per cm, or 81%.

The obvious next choice of model is of the form $k_i = \alpha k_{i-1} + \beta q_i + \gamma q_{i-1} + \delta$ where $\alpha, \beta, \gamma,$ and $\delta$ are constant parameters. This format is common in hydrology, and models of similar type are used in many models involving flow, such as IHACRES ([43]). With a resolution of half an hour, best fit is achieved with $\alpha = 1, \beta = 0, \gamma = 0, \delta = 0$, or in other words $k_i = k_{i-1}$.

Resampling to a resolution of 24 hours and re-fitting the model yields $k_i = 0.94k_{i-1} - 0.14q_i + 0.12q_{i-1} + 13$, but this model is still following the observed $k$ with a one-timestep lag. Consequently, error is highly correlated with gradient to the left of $k_i$ and the model tells us little about the system we could not have found by inspection of Figure 6.1. An inverse relationship on $q$ may be more appropriate, but inspection of Figures 6.1 and 6.2 shows that this model, while it may be more appropriate than the Naive approach we took above, is unlikely to provide a sufficient model.

In the remainder of this paper we develop an alternative to such classical time series modelling. After developing an appropriate data structure in Section 6.3, we use data mining techniques to build a model explaining the behaviour of $k$ and $q$ in Sections 6.4
6.3 Data Structure

Our first task is to build a compact representation of the data. The fine temporal resolution of the record is an asset, but as the difference between adjacent points is so small, there is redundancy in the 31341 point series. A simple discretisation method was therefore applied to both $q$ and $k$ series, breaking them into sections of near-constant gradient, by a process akin to piecewise linear approximation. The data are now in the form of a record of points where slope changes significantly. The $k$ or $q$ value of these points is compared to the original dataset in Figure 6.3. The length of the $k$ series is now 231 points and the length of the $q$ series 248 points, a very significant reduction from the original 31341.

Clearly, some loss of information occurs in deriving this compact representation, but it captures the essential behaviour of both flow and electrical conductivity series very well. Almost no difference between raw and new $q$ and $k$ representations can be noted in Figure 6.3. Now that we have a simplified but hopefully adequate representation, we proceed to the investigatory stage.
6.4 Rule Extraction

Two factors that may influence the value of any given point in the $k$ series are the antecedent electrical conductivity, and the current flow behaviour. For the flow series, the only relationship available for scrutiny is that with antecedent flow conditions, as the prior assumption that $k$ does not affect $q$ is sufficiently fundamental to the above challenge. A rainfall record could be added to the analysis at this point, but a complete record is unavailable.

Assuming that we know little else about the system, a simple way of stating our aim is to say that if the flow behaves as some pattern $Q$, the behaviour pattern $K$ of the electrical conductivity should become known. Here the capitalised letters indicate some event in the flow or electrical conductivity series. This characterisation of the problem leads naturally towards rule extraction methods. Recent applications of temporal data mining techniques for a diverse range of problems include [23], [147], [159]. A discussion of temporal rule extraction as a method can be found in [6]. As noted there, the basic rule $A \Rightarrow B$ will need to be extended to $A \Rightarrow^T B$, meaning if $A$ happens in the flow series, $B$ will follow within time $T$. $B$ may belong to either the $q$ or $k$ series. We shall apply a rule extraction algorithm to each series, and use the output frequent subsequences as archetypes to describe the events that make up the record. Then, the connection between
6.4. RULE EXTRACTION

$q$ and $k$ can be investigated in a simple way.

Complications arise when considering real-valued series and rule extraction, as the technique is usually applied to nominal or discrete data. Our sections of constant gradient, however, are easily divisible into three groups: positive gradient, negative gradient, and zero or near-zero gradient. These categories shall be denoted by the letters U for up, D for down, and 0 zero (flat). Flat sections are defined as those having gradient of magnitude less than 10% of the mean absolute gradient. Visual inspection reveals that this choice of threshold appears reasonable.

The magnitude of the rise or fall and the length of the sections shall be quantified later. To create instances for processing with a rule extraction routine, we use a sliding window that selects a subsequence of the input series. All non-flat points are allowed to act as the head of a subsequence (this follows from the assumption that flatness represents a ground state). The subsequence is cut off when one of the following conditions is met: 1. time between points exceeds a threshold $t_{lim}$, 2. the number of points exceeds a threshold $l_{lim}$, or 3. the system returns to the ground (flat) state. After some trial and error the threshold magnitude, below which gradient is considered flat, was set to 0.05.

The classic Apriori algorithm [3], which is known to be fast and robust, was modified to suit our purposes. Chief among the changes was the preservation of order within the instances, so that [U D] is distinct from [D U]. Apriori extracts frequent itemsets (or in this case subsequences) using the property that no subgroup of a candidate set $C$ can occur frequently if $C$ itself is not frequent, and therefore only the frequent candidates need to be searched for more complex frequent patterns. Frequency can be defined as having a support $s$ (or rate of occurrence) greater than a chosen threshold. Support is most often simply the fraction of the total data pool where the given pattern occurs.

For example, if we have a set of items [milk, cheese, tofu, coffee] from which purchases may be made, and only milk, coffee, and cheese have a support greater than the threshold, no combination that contains tofu can be frequent. From the reduced pool [milk, cheese, coffee], we search the more complex patterns. Of these, perhaps only milk and coffee will be frequent. Thus we could state that if a person buys milk, it is likely that they will also buy coffee and vice-versa. We shall ignore which item or items are the predicate (cause) and which the consequent (effect) in the rule relationship and simply find the patterns that occur frequently. Another relevant quantity is confidence $c$ of a rule (see Section 6.3), here defined as the ratio of the number of instances where the sequence of length $l$ is complete to the total number of joint occurrences of the first $l - 1$
parts of the sequence.

In addressing the issue of capturing gradient magnitudes it is evident that a wide range of possible values exist. Rather than discretise the likely range of gradients, we note that for shape characterisation purposes another distinction exists. Where a string of rising or falling sections occur together, convexity or concavity of subsequences can be established simply by enquiring whether each gradient is larger or smaller than the last. Each instance with a run of U or D can be treated by appending a marker to all but the first value that records whether it is greater or lesser in magnitude than the previous value. Of course, no pair of adjacent gradients will be the same. The notation shall take the form of an appended minus if the magnitude of the gradient reduces, or a subscripted plus otherwise. By the Apriori property ([3]), no convex, concave, or mixed profile may be frequent unless the basic sequence of U, D, and 0 values is also frequent. Therefore, the comparison between adjacent gradients need be made only rarely.

6.5 Frequent Subsequences

Initially, the time threshold \( t_{lim} \) was set at 24 hours for both \( q \) and \( k \). After experimentation it was increased to three days for \( k \), because only small numbers of events with more than two elements were being discovered. There was little difference in the generated dataset when \( t_{lim} \) was increased to four days, and for the \( q \) series the same applied when \( t_{lim} \) was increased to two days. The upper limit on the number of elements \( l_{lim} \) was generously defined as 10. With those settings, plenty of interesting candidate subsequences were generated. It is possible that a characteristic length could have been extracted from the data and used to generate an estimate for \( t_{lim} \), and in further work investigations will be made in this direction. For the moment, however, the subjective trial and error method is deemed to suffice.

The minimum support for the modified Apriori algorithm was set at \( s_{min} = 0.10 \). Note that this was calculated as the proportion of the dataset accounted for by the subsequence divided by the total number of elements, not as the number of occurrences of the subsequence divided by the total number of points. In addition, the frequent subsequences were treated as rules with the last value on the right as consequent. Confidences were calculated for the rules and any rule with confidence less than 0.3 (a rough and naive estimate of chance) deleted. The remaining frequent patterns are listed below, along with support values listed as percentages.
6.5. *Frequent Subsequences*

For $q$:

$$ [U \, D] \ s = 28 \quad [D \, D-] \ s = 15 \quad [U \, D \, D-] \ s = 12 $$

For $k$:

$$ [D \, U] \ s = 21 \quad [U \, U-] \ s = 14 \quad [D \, D-] \ s = 11 \quad [D \, U \, U-] \ s = 10 $$

In $q$, we can see that the first two frequent patterns are subpatterns of the last, although the long pattern does not occur as frequently, either because $[U \, D]$ ends with a 0 value or is interrupted by another event. The distinction is physically important, so minor changes were made to the algorithm so that the confidences for sequences constructed by taking the frequent patterns and appending a zero were also calculated. In this way, a measure of how often the sequence terminates as opposed to being followed by another event was obtained. The probability of $[U \, D]$ being followed by a zero was only 0.35, which rose to 0.76 for $[D \, D \, 0]$ and 0.75 for $[U \, D \, D \, 0]$. So most of the time $[U \, D]$ was followed by another action (other than a zero), which we would expect as $[U \, D \, D]$ is also frequent. From this we can conclude that while $[U \, D]$ is a part of the longer $[U \, D \, D]$, it also terminates in a zero value independently, without being interrupted. In contrast, the support for $[U \, D \, D]$ is only three percent less than the support for $[D \, D]$, and it is fair to conclude that $[D \, D]$ does not usually occur except where $[U \, D \, D]$ occurs.

Convexity/concavity requirements are almost never violated. Therefore the support values effectively remain the same between the raw output from the rule learner and the post-processing convexity check. An exception occurs for the sequence $[D \, D]$ in the $k$ dataset where profiles were divided almost evenly between concave and convex. Apart from $[D \, D]$, the frequent sequences are again all subsequences of the longest frequent pattern. The pattern $[D \, U]$ terminates in a zero with probability 0.46, and $[D \, U \, U]$ has confidence (see Section 2.5.1) $c = 0.49$, so the probability of $[D \, U]$ being an interrupted $[D \, U \, D]$ is very low. Both $[U \, U]$ and $[D \, U \, U]$ have greater than 0.5 probability of being followed by a zero. The level of support for $[U \, U]$ is quite close to that of $[D \, U \, U]$, implying that most of the occurrences of $[U \, U]$ are a part of a $[D \, U \, U]$ pattern.

While the two-part patterns are not necessarily interrupted three-part patterns, the difference between the two may arise through the method used to find sections of constant or near-constant gradient. It is possible that an additional distinction exists within one section of a two-part sequence, but was too fine to be captured. In summary, for the $q$ series we see two characteristic peak shapes: $[U \, D]$ and $[U \, D \, D-]$. Similarly for $k$, $[D \, U]$ and $[D \, U \, U-]$ are characteristic trough shapes.

Before moving onto consideration of the quantitative description of these shapes, we shall consider how much of each series is covered by them. Support values tell us the percentage of the time each sequence is active, but some fraction of each series is settled in
the ground state (zero gradient). Roughly two fifths for $k$ and one quarter for $q$ are flat in terms of number of sections, and considerably more in terms of time. With this data and $s$ values for the characteristic shapes, it can be calculated that the two peak shapes together cover just over 40% of the non-zero $q$ series, and the two trough shapes about 55% of the non-zero $k$ series.

### 6.6 Quantifying Characteristic Shapes

Now that we have a preliminary idea of the shapes that typically occur in each series, it is necessary to quantify them in some way. The left side of Figure 6.4 is a three dimensional plot of flow peaks following the pattern [U D D-]. The axes are the breakpoints $i$ evenly spaced and normalised to begin at the same point (a substitute for a time axis), peak magnitude from initial point to maximum, and the flow value again normalised to be initially zero. The magnitude axis is included to show that while the peak shape does not remain the same over all scales (ie peaks of different magnitude are not congruent) it can be parameterised by magnitude. Therefore a single magnitude value uniquely, albeit approximately, defines peak shape. Similar trends exist for the other frequent subsequences (frequent shapes), but for brevity we shall confine discussion to the [U D D-] pattern.

Consider the height of the shape at each of the defining points as $q_i$, $i = 1, 2, 3, 4$ where the index identifies the point on the time axis. Flow units remain as Litres per second $Ls^{-1}$. We shall assume that each shape has been normalised so that $q_1 = 0$. Similarly, let $t_i$ denote the normalised time at these points. Modelling each of the other $q_i$ with the magnitude $(mag)$ of the peak, which is equal to $q_2$, least squares fitting of degree one yields the follow-
6.7. TWO-DIMENSIONAL RELATIONSHIPS

The time scales of each shape on the left side of Figure 6.4 have been deformed, as the component sections are defined by constant gradient rather than cut off after a set time interval. When proper time scales are added, the plot becomes far more complicated, and the neat parameterisation by magnitude is no longer obvious. However, the length in time of each segment can be parameterised in terms of one quantity. The right side of Figure 6.4 is a plot of break of gradient times \( t_i \) against the total time \( t_{tot} \). All times are normalised to begin at zero, so \( t_1 \) is uniformly zero and so is not shown. The times \( t_2 \) where the first section ends and \( t_3 \) where the second section ends can be roughly modelled as a function of the total time. In the case of [U D D-] total time \( t_{tot} = t_4 \). Least squares linear models for each unknown \( t_i \) are given below, along with the mean error in half-hour time periods and hours.

\[
\begin{align*}
q_3 &= 0.28 \text{mag} + 0.91 \quad \text{with mean error} = 17Ls^{-1} \\
q_4 &= 0.13 \text{mag} - 10 \quad \text{with mean error} = 17Ls^{-1}
\end{align*}
\]

\[
\begin{align*}
t_2 &= 0.086t_{tot} + 7.719 \quad \text{with mean error} = 7.3 \text{ (3.7 hours)} \\
\quad t_3 &= 0.316t_{tot} + 13.002 \quad \text{with mean error} = 13.0 \text{ (6.5 hours)}
\end{align*}
\]

Inspection of the right-hand side of Figure 6.4 suggests that \( t_2 \) and \( t_3 \) could also be reasonably estimated with by constants. Mean errors from the constant approximation were 4.0 and 8.5 hours for \( t_2 \) and \( t_3 \) respectively, a significant rise particularly in the former case.

Similar parameterisations can be found for all the other frequently occurring peak and trough shapes. We now have all the necessary time and flow magnitude information representing a [U D D-] peak encoded in two numbers, \( \text{mag} \) and \( t_{tot} \). Of course, to recreate the time series we do still need to know the position of the peak in time and the starting flow value. Investigation into a relationship between those two variables failed to reveal any useful information, as a spectrum of total times exist for any given flow event magnitude.

6.7 Two-dimensional Relationships

Now that characteristics of each series individually have been investigated, we can return to the original problem of discovering the relationship between flow and electrical conductivity. If a large number of different characteristic shapes had been identified as frequent in either
series, we could proceed by building new instances for rule learning where each $q$ event was followed by any $k$ event that occurred within a certain time. However, the behaviour of each series has been decomposed into only two main patterns, and we shall simply test the hypothesis that a peak in flow causes either a [D U] or [D U U-] event in $k$.

Setting the time limit between events at a generous two days between the start of the flow peak and the start of the $k$ trough, we find 22 occurrences of a trough close to a peak. Recalling support values, we find that 34 troughs and 35 peaks were identified. Some of these were small in size, and examination of the magnitude values of generating and generated shapes reveals that they are mostly relatively large. Small peaks do not typically cause detectable changes in $k$ and, correspondingly, many small $k$ events cannot be traced back to an event in $q$.

Curiously, in a handful of cases the trough actually appeared to begin before the peak. These points were checked in detail, and there often appeared to be no other possible generative peak in streamflow to the left. Therefore we allow the condition, that the $k$ response follow the $q$ event, be relaxed to simple occurrence within an interval around the start of the peak. This behaviour may seem unphysical, but it is in keeping with our policy of making as few assumptions as possible. Most of the troughs, however, fall within 12 hours after the streamflow peak begins.

In the last few months of the record, several events are recorded in $k$, but no matching $q$ peaks were identified. This anomalous behaviour can be observed by very close inspection of Figure 6.1 as well, although it is by no means immediately apparent, even at high magnification. The researchers originally responsible for collecting the data revealed that the flow record in this region was in fact questionable, due to the gradual degradation of the streamflow recording mechanism [140]. This resolves the inconsistency in the hypothesis that a $q$ event implies a $k$ event.

Inspection reveals that while there is a mild tendency towards [U D] peaks that are not subsequences of [U D D] producing troughs of the shorter form, and similarly long peaks [U D D-] producing the longer form of trough, it is not at all pronounced. Support information yielded the observation that while some of the two-part patterns are the beginnings of three-part patterns, this is not always true. Some may be interrupted, but others terminate. Ideally, all three cases would be treated separately along with the three-part pattern, but analysis at this level of detail demands more data than we have available.

The problem of finding a relationship between magnitude of events in the two variables remains. Simply plotting the two against one another yields very little information. How-
ever, when the initial state of $k$ is introduced into the plot as an additional axis, useful order begins to appear. Figure 6.5 shows one view of a plot with magnitude of flow peak $q_{mag}$, initial $k$ state $k_{state}$, and the minimum point $k_{min}$ to which $k$ drops. When combined with $k_{state}$, this last variable contains the same information as the magnitude of the trough in $k$, but produces a clearer trend which suggests a surface on which $k_{min}$ is proportional to $k_{state}$ with gradient decreasing with increasing $q_{mag}$. While this two-dimensional view is not as informative as the rotatable three-dimensional plot, the relationship is quite pronounced under even this visualisation. It is quite surprising, especially the relatively weak coupling between $q_{mag}$ and $k_{mag}$. However, it is useful. In the dense region $q_{mag} < 200$ or so, the relationship between $k_{state}$ and $k_{min}$ is fairly well defined, but becomes a matter of hypothesis in the sparser region where fewer events have been observed.

The simple hypothesis that the total time of the $k$ event increases with the total time of the $q$ event can be proven false by a simple scatter plot (Figure 6.6). Note from this plot that there appears to be a characteristic time scale for $q$ events, but a wide variation in total times for $k$ events. Clearly, other factors are at work. Experiments with multivariate linear regression did not reveal a sufficiently strong relationship with either the $q$ time parameter (total flow event time) or any other variable to determine time scale of the $k$ event.
6.8 Conclusions

This exploration of the $q$ and $k$ series of record from Hollin Cave and the relationship between them revealed strong regularities in both series. Characteristic peak and trough shapes, easily identifiable by cursory inspection of Figure 6.1, were extracted by the modified Apriori algorithm. It also revealed that the two-part patterns [U D] and [D U] existed independently from the subsequences [U D D] and [D U U], although the difference between the short and long patterns may be an artefact of the method used to break the series into lengths of near-homogenous gradient. In both $q$ and $k$, the three-part sequences proved to be of almost uniform convexity and were thus refined to [U D D-] and [D U U-], thus removing unusual and probably unhelpful data from the shape analysis. The examination of shape convexity is to the author’s knowledge novel, and is expected to be useful in the future analysis of larger and more unwieldy datasets. Figure 6.4 illustrates the [U D D-] patterns, and shows that the peaks are not congruent to one another, but can be parameterised by magnitude and total time. Similar results were obtained for the other frequently occurring shapes.

The hypothesis that events were linked in time was then verified. A relationship between flow peak size, initial $k$ state and the point to which $k$ falls was established for the dense region $q < 200 \, \text{Ls}^{-1}$ and postulated for $q \geq 200 \, \text{Ls}^{-1}$. The initial state of the conductivity at the start of a trough, $k_{\text{state}}$, proved to be far more important than the size of the
6.8. CONCLUSIONS

precipitating flow event. The problem of determining the time scale of \( k \) events remains, although we have established that flow peak magnitude does not have a significant effect, and there is some reliance on \( k_{\text{min}} \), the minimum conductivity attained. When the nature of total event time \( t_{\text{tot}} \) can be established for \( k \), a complete model will have been obtained and we will be able to reconstruct both time series. Given a starting value of \( k \), the series may be recoverable to a reasonable accuracy.

In terms of the dynamics of Hollin Creek system, the fact that electrical conductivity on some occasions begins to fall before any increase in flow may be due to the fact that Hollin Creek is certainly connected to another system underground through complex avenues. Unlike most surface streams, the Hollin Creek flows through restricted spaces, and some passages may only be used at high flows. When presented with the above results, speleological experts [140] suggested that as water levels in the cave begin to rise, a branch of Hollin creek takes a different path through the cave, collecting less salts or allowing more to be precipitated out as calcium carbonate (calcite). This could result in a small quantity of very low conductivity water entering the system before the main peak in flow, as we see.

It has also been suggested that the flow peaks are delayed in some way by cave passages too small to accommodate the full flow. The system may contain a bottleneck where water accumulates until the water level rises to a second passage where it can overflow. Complex three dimensional structures of this kind are common in many caves. Either or both of these dynamics may be at work in Hollin Cave, and may also explain why a stable relationship between \( q \) and \( k \) time scales remains so elusive.

These two points form the main focus for work on this problem in the near future. Other approaches to the time issue, such as uniform sampling, are also under consideration. The methods developed will also be applied to other, longer records, and datasets with a greater number of interconnected variables. Unfortunately, the short length of record prevents us from testing on an independent sample and so we cannot obtain real measures of how well the system dynamics can be reproduced, but the model seems to reproduce some aspects of the flow/electrical conductivity system dynamics well, but it is clear from Figure 6.5 that real system knowledge has been gained.

Although we do not yet have a complete picture of the dynamics at work in the flow/electrical conductivity system, our qualitative and quantitative knowledge has been advanced considerably over that discussed in the introduction and in Section 6.2, where we indicated that a peak in streamflow seemed to cause a sharp drop in electrical conductivity followed by a slow recovery. Qualitatively, both series are made of characteristic peak and
trough events. The depth of the troughs in $k$ is determined largely by the starting point but also by the magnitude of the corresponding $q$ peak. Quantitatively, the relations controlling event shapes have been extracted, and for most flow magnitudes the relationship between peak and trough magnitudes can be determined. The nature of this relationship is unexpected, with surprisingly weak dependence on flow peak magnitude for most flows, and correspondingly strong dependence on the initial state of $k$. 
Chapter 7

Clustering Daily Flow Time Series

7.1 Introduction

As noted in Chapter 1, a great deal of environmental data is presented in the form of time series. These time series may be records of streamflow, temperature, rainfall, chemical concentration in water or air, animal populations, tree heights, or any number of other variables. Often, time series of one variable are recorded at multiple sites. For example, consider the water quality measurements taken from Jugiong, Tarcutta, and Muttama catchments analysed in Chapter 5. That dataset contains only three sites, but in this chapter we shall use a collection of 36 daily streamflow records from South-East Queensland. The size of this database is large enough to make manual series clustering impractical, and to encompass a range of stream characteristics.

There can be several aims in clustering such data. Processing each individual series may be prohibitively a time-consuming activity, so it would be useful to split the dataset into groups of similar series, so that only one or two from each group (or cluster) would need to be analysed in detail. Alternately, scenarios may exist where, for example, a certain management scheme should be applied to all regions with characteristics similar to a given archetype, although these are not known a priori. It is also useful to be able to automatically identify series with similar characteristics for regionalisation purposes - where the parameters of one model are applied in another region where behaviour is known to be similar, but where sufficient data for model calibration does not exist. Regionalisation is an active research field in hydrology, and is attracting increased attention under new co-operative research projects such as the Prediction in Ungauged Basins (PUB) initiative (see for example [132]), and clustering by data mining methods like that used below is an
efficient way of extracting and presenting information on similar and dissimilar series.

There are many reasons that natural groupings within a dataset may be useful. The purpose of this chapter is to investigate and illustrate the different ways time series similarity, and hence time series clustering, can be approached. As we shall see, choice of distance measure is central to the clustering problem, and must be related to the ultimate goal of the analysis. Cluster stability will also be considered as an issue here, as we consider it important that clustering schemes should be accompanied by such information.

The dataset we shall first use to consider the problem of clustering time series consists of 36 daily flow records from catchments in the Brisbane area of Queensland, Australia. This data is available through the Australian Bureau of Meteorology, and was selected because it is representative of normal quality, length, and characteristics of Australian streamflow datasets. A range of flow schemes are represented, for example with respect to baseflow and response times. The data is discussed further in 7.3, and in [24].

As noted in Sections 2.3 and 2.6, the definition of a distance measure is the key to clustering time series. While a 'true' or 'correct' clustering scheme has been defined for most of the datasets used to build and compare distance measures, this is not universally true, particularly in the environment sciences. A cursory investigation of the flow time series used throughout this chapter, for example in Figure 7.8 shows that no absolute 'correct' clustering scheme suggests itself to the human eye. Therefore, we cannot define a quantitative performance measure to tell us how good or bad the results of a particular clustering method are. Instead, we must make a careful and informed choice of method based on the question we wish to answer.

According to the specific problem at hand, we may want to classify rivers and streams according to baseflow, response time and peak width ('flashiness'), geographical location, or discover groups with corresponding peaks and troughs. Classification techniques can do all or any of these things, when an appropriate distance measure is defined. In this chapter, the Euclidean and total absolute difference are applied as distance measures, as well as the pointwise mutual information (see Section 7.4.1) between discretised series and a compression-based measure derived from the shape-based investigation of Chapter 6.

We do not intend to provide a review of distance measures and clustering algorithms, as this has been done with far greater depth than we could hope to achieve here (for example see [75], where it is noted that over 200 distance measures have been published). Tables of statistical performance of various clustering schemes are easy to come by, and are for the most part calculated on clean data, ie that with low uncertainty, few or no missing values,
and no outliers. Rather, this chapter aims to illustrate the utility of different distance measures and demonstrate the creation of a new measure for a specific task. Along the way difficulties with missing data points and uncertain problem definition are encountered and addressed. While the dataset used here is a real-world collection and the clusters we obtain are physically meaningful, no physical interpretation shall be attempted. In the next chapter, we build upon the techniques developed here to examine a large chemical trace dataset.

7.2 Distance Measures and Metrics

Aside from clustering, similarity between two (or more) time series is often used to quantitatively measure the difference between output of a model and observed system behaviour. Euclidean distance or the sum of absolute pointwise errors is often used for this purpose, as is efficiency

$$R^2 = 1 - \frac{\sum (x_{\text{observed}} - x_{\text{modelled}})^2}{\sum (x_{\text{observed}} - \text{mean}(x_{\text{observed}}))^2}$$

([100]), bias (ratio of area under modelled curve to area under observed curve), any other measure or some combination of these, or an objective function defined specifically for the problem at hand. Nearest neighbour methods (see Section 2.4.8) also require a distance measure of some sort, so that near neighbours and their weights can be extracted. Similarity can be thought of as the inverse of distance, but either concept requires a quantitative description.

Clustering is discussed in Chapter 2. As noted in [81], more than a thousand papers have been published on the subject of time series clustering, using more than 200 different distance measures ([75]), but surprisingly little headway has been made in this area. As noted there, most methods have been designed for and tested on only one or two datasets, often of similar type. For examples of recent research in the area, see [10], [72], [79], [76], the many references given therein, and many others. The very extensive survey performed in [75] investigates, re-implements, and tests a large number of these.

Most distance measures are defined such that a number of useful properties hold, namely the following:

$$d(A, B) \geq 0$$
Figure 7.1: Illustration of the Triangle Inequality

\[ d(A, A) = 0 \]

\[ d(A, B) = 0 \text{ if } A = B \]

\[ d(A, E) = d(B, A) \]

\[ d(A, B) + d(B, C) \geq d(A, C) \]

The first four are largely self-explanatory: that no distance may be negative and the only distance with zero magnitude is that between identical quantities, and that the distance measure is symmetric. The action of the fifth condition, that of the triangle inequality, is illustrated in Figure 7.1. Clearly, in a well-defined space \( d(A, C) \leq d(A, B) + d(B, C) \). Under these constraints, a broad range of distance measures can be defined. These are called distance metrics.

Most distance measures for time series fall into two categories: point-wise, exemplified by Euclidean distance, and those that consider characteristics over whole series, such as comparison between Fourier spectra. The latter are sometimes known as compression-based, because they rely on a reduced representation of the series. Both types of similarity measure are applied below. It is worth noting that the mutual information measure discussed in Section 7.4.1 does not conform to all the properties listed above and is hence not a metric. In particular, note that \( d(A, B) = 0 \) in cases other than \( A = B \), and the triangle inequality is not satisfied.
7.3. Basic Investigation

The gauges generating the 36 streamflow series are all located in South-East Queensland, Australia, in or near the Brisbane River catchment. All measurements are daily in resolution, although significant missing periods exist in many of the series. The data is publicly available from the Australian Bureau of Meteorology (see [2]). The earliest point in the dataset is 9 August 1963 and the last 3 June 1999, although the exact start and end dates of each series are different. For simplicity, we shall consider all series as beginning on 9 August 1963 and ending on 3 June 1999, and flag the points without measurements as missing data. In this way, we obtain 36 series \( X_i, i = 1, \ldots, 36 \) of 13399 points each, although the amount of useful data is less than this.

Using Tobler’s Law ([95]), the adage much used in geographical and environmental sciences, that all things are related but near things are more closely related than far things, we shall begin by ignoring the series themselves and cluster by physical location only. As the results of such a routine are visually verifiable, this experiment shall also serve as a test of the cluster-generating code.

Probably the most popular clustering technique is the k-means algorithm (see Section 2.3). The convenient simplicity of the k-means method is countered by the fact that the clusters generated are locally optimal, but not necessarily globally optimal. To get around this, one can perform multiple runs of the algorithm using different random seeds (initial cluster centres) each time, to test for stability as we did in Chapter 4. Of course, the choice of \( k \) (number of clusters) must also be made a priori. Rather than perform detailed tests to determine appropriate \( k \), we shall set the parameter higher than required and allow clusters to become empty if the cluster centre is the only occupant.

7.3.1 Stability Considerations and Geographical Clustering

Cluster stability is an important issue to consider, and must be validated like any other model describing data. Low stability may imply a poor choice of distance measure or clustering algorithm, or simply that clusters are not well separated or that natural groupings do not in fact exist. In the case of the South-East Queensland flow data, we know that no strong clustering scheme dominates all other possibilities, so we shall instead look for trends in the clustering scheme and test our method on the simplest case - geographical clustering based on Euclidean distance between gauge locations.

We shall run the algorithm 100 times with \( k = 10 \), at each iteration incrementing a
matrix of co-occurrences $C$ to keep track of associations. The low computational cost of the k-means algorithm makes this a practical experiment. For each pair of flow series $X_i$ and $X_j$, $C(i, j)$ is incremented when $X_i$ and $X_j$ are placed in the same cluster. Using this matrix, we can extract those series usually grouped together. Using the L1 distance $\frac{1}{N} \sum_{i=1}^{N} |x_i - y_i|$ between flow gauge locations and the method outlined above, we arrive at a matrix $C$ shown in raster form in Figure 7.2.

The red regions indicate a high rate of co-occurrence, shading to blue, where co-occurrence is low. This is the colour scale we shall use throughout this chapter, with dark red at the maximum and dark blue at the minimum. The fact that the order of the series is geographically motivated makes this figure easy to read - as noted earlier, geographically close streams are in general more likely to be similar. Clearly, series one through nine regularly form a cluster, as do 11, 12, 13, 14, 26, 27, 28, 29, and 18, 23, 24, 25, and also 33, 34, 35. The large block centred around $i = 20$ also indicates a cluster containing 19, 20, 21 and perhaps others.

To automate and formalise the membership of clusters we shall state that all $X_i$ belongs to the same cluster as $X_j$ if the entry $C(i, j) > 50\%$ of the total number of runs (50 out of 100). Note that only one value of $j$ is required to pull a new member into the cluster, and where conflicts arise between which cluster a series should best be assigned, they are resolved arbitrarily. While this is clearly not a perfect system, it is a quick way to calculate cluster membership and appears to work well.

The threshold on $C(i, j)$ for cluster membership assignment, which we shall refer to as $C_{thr}$, can be varied to obtain tighter clusters if required. Using this method, the clusters
Figure 7.3: Gauge Locations Coloured by Geographical Cluster

shown in Figure 7.2 are as follows. Note that certain series, such as number 10, do not appear to consistently belong to any particular cluster.

1 1,2,3,4,5,6,7,8,9
2 18,19,20,21,23,24,25
3 16,17,22
4 11,12,13,14,26,27,28,29,30,31,32
5 15,33,34,35,36

Figure 7.3 shows the Universal Transverse Mercator (UTM) co-ordinates of flow gauges coloured by cluster membership. By visual inspection they are qualitatively reasonable, and one can see that if the number of clusters were to be tightened, the red and green clusters would likely aggregate, as would the yellow and purple.

The next obvious test to make is to calculate L1 and Euclidean distances between the series themselves. Before doing this, we may choose to normalise the series or not. Where normalisation is performed, series are translated to have a minimum of zero and then scaled to have a a maximum of unity. Also, because the series have varying endpoints the overlapping period of each pair of series is not necessarily the same. Therefore the total L1 distance is applied to all common points only, becoming an average absolute distance
between all overlapping points in series \(i\) and \(j\). We encounter a problem at this point—a few pairs have no common points at all. While all series contain several years of data, the time periods covered sometimes do not overlap. To deal with this, a Missing Distance Estimate (MDE) method was developed and tested.

### 7.3.2 Missing Distance Estimation

In some way, we need to fill the gaps in the matrix of distances \(D\). Using the assumption that the dataset is reasonably dense in whatever space we are considering (i.e., that any given series will not be completely isolated), the distance \(D(i, j)\) between series \(i\) and series \(j\) was replaced with the mean of \(D(i, a)\) and \(D(b, j)\), where \(a = \text{argmin}_m D(i, m), m \neq j\) \(b = \text{argmin}_m D(m, j), m \neq i\). In other words, instead of calculating \(D(i, j)\), the distance between \(X_j\) and the series nearest to \(X_i\) and the distance between \(X_i\) and the series most like \(X_j\) are averaged and used as an estimate.

Consider again the triangle inequality and Figure 7.1. We are assuming that the distance \(d(A, B)\) is small where \(B\) are the closest series to \(A\), and hence \(d(A, C)\) can be approximated by \(d(B, C)\).

Appropriateness of this estimation method was tested with the 11 distances between gauge locations. To generate estimates, a number \(n_{\text{miss}}\) of \(i, j\) pairs were chosen at random from the distance matrix and the corresponding \(D(i, j)\) and \(D(j, i)\) replaced with missing data markers. The new \(D\) was then fed through the estimation algorithm. With \(n_{\text{miss}} = 10\) distances, the average absolute percentage error from 50 repetitions of the experiment was 17.8\% and mean absolute error 1.28 km. Figure 7.4 shows the distance estimates plotted against the original values.

As a benchmark, the error generated using the average distance between all possible \(i, j\) pairs (subject to the condition that \(i \neq j\)) was also calculated and used as a replacement for all missing distances. The average errors in this case were 192\% and 4.70 km respectively.

When the number of missing distances is doubled, the error rates rose to 33.9\% and 1.71 km. These values remain much smaller than the benchmarks. When \(n_{\text{miss}}\) is halved, they fall to 9.22\% and 0.97 km. The reason for the variable performance is that as the number of missing entries in \(D\) rises, the best entries for replacing any given entry have a higher probability of being missing themselves, and hence we must look further. We conclude that the method is good for replacing missing distances where \(D\) is mostly whole.

Note that in the case of the two-dimensional distance measure used to generate \(D\) in this experiment, missing \(D(i, j)\) could be replaced analytically. However, this is not
Figure 7.4: Missing Distance Estimates from MDE Algorithm and L1 Distance Matrix possible for the more complex distance measures used to construct $D$ matrices later in this chapter, so we shall use the method described above.

7.4 Distances between Series

To avoid flooding the MDE algorithm with too many missing values and also to decrease processing times, a three year sub-period was chosen from the 1963-1999 span of the dataset, and only the records covering this period were clustered. In this way, we arrive at 21 flow series of 1096 points each for the period 1 January 1992 to 31 December 1994. This shall be called the 'reduced' dataset, and holds the most complete overlapping set of series.

Clusters generated with the L1 distance calculated between series are illustrated in Figure 7.5 for raw and normalised cases. The two pictures are noticeably different, which illustrates the difference that choice of pre-processing method can make.

The blue cluster down the right side of the image remains the same for normalised and un-normalised cases, but the membership of other clusters varies slightly. Were our application to require differentiation between high and low baseflow series, or peak magnitudes, we would most likely choose not to normalise the data before clustering, but if we were more concerned with matching the timing of flow peaks, normalisation would be
CHAPTER 7. CLUSTERING TIME SERIES

Figure 7.5: Gauge Locations Coloured by Series L1 Cluster, for Raw and Normalised Data recommended.

7.4.1 Mutual Information as Similarity Measure

Recall that in Chapters 5 and 6, it was useful to consider discretised series. In that way, information was both codified for addition to the analysis, and expressed in an interpretable fashion. It has been found that severely discretised series can be clustered as meaningfully as the original data (see [10]) in some cases. The discretisation in the case was hard clipping, where data values are transformed to 0 if they lie below the mean and 1 if above. Here, we shall consider the first/second order difference scheme used in Chapter 6, which we know to be useful in the context of flow data.

There are several ways a distance measure can be defined over discrete sequences. The method preferred by [80] is the definition of a lookup table based on a particular symbolic representation. The very rigorous compression-based measure discussed therein is compared with the Euclidean in the same paper. Others (for example [12] and [19]) use variants of the Shannon's entropy $H$ (see for example [87] or [62]):

$$h(x = a_i) = \log_2 \frac{1}{p_i}$$

$$H(X) = -\sum_i p_i \log_2 p_i$$
Here $p_i$ is the prior probability of the outcome $x = a_i$, with $a_i \in A$. Note that the entropy $H$ is merely an average Shannon information content $h$ over the whole of $X$. At this point, it is instructive to note the use of the information gain measure $I$ (Equation 2.2) in Chapter 2. While our language here is rather more formal, the concepts under discussion are exactly the same.

Note $0 \leq H(X) \leq \log(N)$ where $N$ is the total number of classes. The minimum value occurs where every entry in $X$ is identical, and the maximum when $p_i = \frac{1}{N}$ $\forall i$. This is consistent with the fact that the outcome of a random experiment is guaranteed to be most informative if the probability distribution is uniform ([87], page 70), which can be demonstrated with a simple, intuitive game. In order to find the value of a number between one and 16, we may ask only questions such as 'is it greater than 7?'. To determine the number in the least number of questions, the logical process is to ask 'is it greater than 8?' first, dividing the set of possible outcomes into two parts. The target number is then isolated in one half of the set of possible values. Then, depending on the answer, the next question should be 'is it greater than 12?' or 'is it greater than 4?', dividing the remaining possibilities into two more equal parts. In this way, we isolate the chosen number in four questions.

This example also illustrates some of the properties of the entropy function: the maximum lies at $h(x = a_i) = H(X) = 4\text{bits}$ when all outcomes are equally probable, and tells us that a maximum of four binary digits are required to encode any number up to 16. In the general case, where the probability distribution may not necessarily be uniform, entropy can be thought of as the number of binary decisions (digits) required to uniquely determine (store) an outcome of the random variable in question.

As noted in [62], $H(X)$ can also be considered as a measure of the effective number of possible outcomes. This may be useful where some values are highly unlikely, and thus do not contribute much to the characteristics of the data. The concept of entropy can be extended to cover datasets with more than one feature. While the formulae given below cover only the case of two related variables, the concepts are easily extendable to higher dimensional problems.

Consider two data vectors $X$ (as above) and $R$ taking values $r = b_j, b_j \in B$, that are related some way. For the present, call $X$ the descriptive feature and $R$ the target variable. A quantification of the amount of information about $R$ that scrutiny of $X$ can provide us with is embodied in the cross entropy $H(R,X)$, also referred to as joint entropy in some literature:
CHAPTER 7. CLUSTERING TIME SERIES

\[ H(R, X) = - \sum_{ij} p_{ij} \log_2 p_{ij} \]

Here, \( p_{ij} \) is the joint probability of observing \( r = b_j \) and \( x = a_i \). If \( R \) and \( X \) are independent, \( H(R, X) = H(R) + H(X) \), but this is rarely the case. See for example [62] for a more detailed discussion. From [87], we take the following further definitions with minimal alteration.

Conditional Entropy of \( R \) given \( X \):

\[ H(R|X) = - \sum_{ij} p_{ij} \log_2 p(r_j|x = a_i) \]

Mutual Information between \( R \) and \( X \):

\[ M(R; X) = H(R) - H(R|X) \]

Regarding these two quantities, [62] makes a few useful observations. Firstly, the conditional entropy measures the extent of behaviour in \( R \) not explained by \( X \). From this, it follows that mutual information between \( R \) and \( X \) is at a maximum when the conditional entropy \( H(R|X) \) is zero, as is clearly the case, and the maximum value of \( M(R; X) \) is the entropy \( H(R) \). The minimum value of the mutual information is zero, as would be expected from a well-behaved function. When mutual information \( M(R; X) \) is at a maximum and thus \( H(R|X) \) is zero, the value of \( X \) completely determines the value of \( R \). In this situation, the descriptive variable \( X \) provides all the information required to determine the target \( R \). Of course, this will rarely if ever happen in practice except when \( R = X \), but the proximity of \( M(R; X) \) to its maximum gives an indication of how close the observed case is to the measured, or the likeness of one variable to another.

Recalling the distance metric conformities set out in Section 7.2, we note that mutual information is a similarity measure (i.e. that it increases with similarity and decreases with distance), but this does not represent any obstacle to clustering. The minimum value of \( M(R; X) \) is zero, and the maximum value varies but is calculable, and therefore we can consider \( \frac{M(R; X)}{\max(M(R; X))} \), which approaches unity when \( R \) and \( X \) are very similar. For the case \( M(R; X) \neq 0 \), we can simply consider the inverse of this fraction and set \( d \) as a very large value when \( M(R; X) = 0 \). Note also that mutual information is symmetric, but it does not satisfy the triangle inequality and is therefore not a metric. Also, as noted above, \( d(A, B) = 0 \) does not strictly imply that \( A = B \). Information theory-based metrics have
been built by others (for an example see [12]), but for our purposes a measure, rather than a metric, will suffice.

In the case where the distribution of values remains constant or roughly constant, \( H(X) \) is constant over all \( X \), and we can make significant simplifications. The maximum mutual information must also remain constant, so we can consider \( d = \frac{1}{M(R;X)} \), \( M(R;X) \neq 0 \). We could go even further and eliminate the constant from the mutual information calculation itself.

Here, as in Chapter 5, segments of the series were recorded as flat (0), rising (1, U), falling (2, D). These last two categories were then further classified as in Chapter 6: where \( Y_i(s) = Y_i(s + 1) \), \( Y_i(s + 1) = Y_i(s + 1) + 1 \) if the magnitude of the rise or fall in position \( s + 1 \) is larger than that of \( s \) and \( Y_i(s + 1) = Y_i(s + 1) + 2 \) if it is lesser. A threshold of 10\% of the mean absolute rise/fall was defined, below which segments were considered and 10\% of the mean absolute difference between adjacent rises/falls was used in a similar way as a minimum difference below which two rising or two falling segments were considered the same. In this way we arrive at the discretisation used in Section 6.4:

0 Flat
1 U rising
2 D falling
3 U+ rising, following a lesser rise
4 U- rising, following a greater rise
5 D+ falling, following a lesser fall
6 D- falling, following a greater fall

Note that when using this scheme it is necessary to take into account the fact that U+ and U- can only occur after U, and similarly D+ and D- after D. Certain pattern such as [0 U+] and [U D-] are disallowed.

Discretisation could also be achieved by hard clipping, dividing the data range into a number of buckets, or any number of other methods. Choice of discretisation method, like distance measure, should be determined by the object of the investigation. Here, we are quantifying how much of the discrete series \( B' \) can be determined by the discrete series \( A' \). In regard to the commonly imposed constraints discussed in Section 7.2, note that the
value of series $A'$ may uniquely determine the value of $B'$ without $A' = B'$. Consider the case where two differently behaving variables, say streamflow and water temperature, are driven by an independent variable such as rainfall. It may be, that a rainfall event causes a sudden rise in flow, but lowers the in-stream temperature as cold water is added.

As illustrated by Figure 7.6, these series are clearly related, but a pointwise distance measure may not reflect this. With a good discretisation scheme, the closeness of the relationship will be captured by the mutual information distance measure. Herein lies its utility.

For the dataset at hand, we must also include another value to mark the position of a missing measurement (-99). However, there are only a few missing values extant in the reduced system of 21 series, and these appear to be distributed near-randomly. Thus, the missing points are merely passed over by the mutual information calculations.

Using the assumption that $H(X)$ is roughly constant for all series $X$, which is allowable as the distribution of discretised values over most of the South-East Queensland flow series are very similar, a matrix $D$ of distances was calculated, such that $D(i, j)$ equals the distance between series $i$ and series $j$. The distance matrix $D$ shall be used throughout this chapter and the next, and was in this case calculated as the inverse of mutual information as distance matrix (or where mutual information $M(i, j)$ approaches zero, a very large constant value). Figure 7.7 shows gauge locations coloured by clusters generated with this matrix and the k-means algorithm. Note that two series were not placed in a cluster with a reasonable degree of certainty. The mutual information between series $X_i$ and $X_j$ was computed using discrete transformed series $Y_i$ and $Y_j$ obtained using the procedure described in Section 6.4.
7.5 Distance Measures between Series

Using (as discussed above) a distance matrix $D$ such that

$$D(i, j) = \frac{1}{M(i, j)}$$

where $M(i, j) > \epsilon$, or

$$D(i, j) = \frac{1}{\epsilon}$$

otherwise, clustering over the set of flow series was performed. With mild experimentation, $\epsilon$ was set to 0.005. In this way, the cluster assignment illustrated by Figure 7.7 was achieved.

Figures 7.3 to 7.7 show that while clusters do vary according to choice of distance measure, a geographical signal does exist. This is not surprising as gauges that are geographically close are likely to be driven by the same rainfall events. If we had chosen to reduce the series to, for example, spectra of some kind or cumulative distribution functions, this time-dependent aspect would be lost. However, from a modelling point of view we are interested in which variables are driven by the same underlying signal.

Figures 7.8 to 7.10 are generated with the cluster assignments obtained from the process outlined above, except that the threshold $Cthr$ on cluster membership has been tightened to 80% from 50%. As throughout this chapter, the time step for these series is daily. The altered threshold was intended to reduce the number of elements that needed to be
represented, and to make the differences between displayed clusters more distinct. The figures show a section of all records assigned a cluster, coloured by cluster number.

Note that the two high baseflow hydrographs (series 10 and 17) shown in red in Figure 7.8 and green in Figure 7.10 are placed together in a cluster of their own by the un-normalised and mutual information measures. These were not assigned to clusters at all when we used the normalised L1 distance, as series 17 was left out of the reduced dataset and series 10 is highly dissimilar to any other series. The two red series in Figures 7.9 and 7.10 are grouped together by the normalised L1 and mutual information measures, but not by the raw L1. Mutual information fails to make any further distinctions at this level, while the other measures break up the rest of the series in different ways.

### 7.6 Hierarchical Clustering

The routines outlined in the above section provide the information required to perform hierarchical clustering in two different ways. The first, standard method would be to
Figure 7.9: Normalised Flow Series (reduced) Coloured by Series L1 Cluster
Figure 7.10: Flow Series (reduced) Coloured by Mutual Information Cluster
group the closest series together, and then add new series to these pairs by estimating the distance between a designated cluster centre and other clusters or unassigned series. In this way a tree or hierarchy of clusters is built up, but distances to the cluster centre (however this is defined) must be estimated for all other items. Therefore, we shall simply vary the threshold $C_{thr}$ from very tight to very loose, thus adding more series to the initial tight clusters. This does not require substantial processing time, as distance matrices already exist.

Note that where $C_{thr} \leq 50\%$, an element may be assigned to more than one cluster under the rules set out above. Therefore, the threshold was varied from high to low, and where ambiguity occurs that element is kept with those it appeared with in the previous iteration. Where no previous association was detected, the series was assigned arbitrarily. Happily, this situation seems to occur rarely in this case. If we were going to use hierarchical clustering in any serious way, stricter rules would have to be set out, but for the purpose of illustration the current method suffices. Figure 7.11 illustrates the hierarchical clustering scheme so generated. This type of visualisation is sometimes called a dendrogram.

Note that when performing hierarchical clustering, we have the opportunity to be more sophisticated in resolving to which cluster a series should be assigned, because we can follow the path of the series through the dendrogram and assign the best cluster based on the cluster membership at tighter count thresholds. Another interesting feature of Figure 7.11 is that many small clusters aggregate at the same point. This could be considered a threshold point, at which a natural clustering scheme is suggested.
7.7 Clustering with Frequent Patterns

As demonstrated in previous sections, different distance measures produce significantly different final clustering regimes, even though those regimes may be quite stable under changes to initial conditions. In practice, the choice of distance measure must be made carefully and with consideration of the problem at hand. Euclidean, Manhattan, and $L^p$ norm distance measures consider only the total distance between pairs of points, normalised or otherwise. The Dynamic Time Warping approach advocated in [75] (and briefly mentioned in Section 2.6.3) allows the time axis to stretch and compress, but is still essentially a pointwise comparison method.

For variables such as streamflow, we may also be interested in correspondence in time with a delay. In larger catchments, there is typically a longer period of time between incident rainfall and the peak of streamflow response. Thus, a single rainfall event may trigger flow peaks at slightly different times in several nearby catchments. A similarity measure which allows for such delays could be established for use in this situation.

The mutual information measure outlined above takes no account of the magnitude of the point to point variations, and cares only that they appear to be driven by the same signal in time. The reliance on correspondence in time is a weakness of this and any other point-to-point distance measure, although they are of course very valuable tools for solving a wide range of problems. Indeed, identifying correspondence in time is often the target of our analysis. However, we sometimes do not have an overlapping time period between our two series, but we would still like to define some measure of similarity. In these cases we must look elsewhere for an appropriate measure, for example to a comparison between Fourier spectra of the series.

Dimensional reduction methods such as Fourier decomposition look only at the overall frequencies of the collection of sine waves that approximate the series, which can then be easily compared between two or more series, and does not care whether events occur in step or not. As demonstrated in Section 2.6.3, good Fourier compression requires a strong periodic signal, which we know by visual inspection exists for some of the series in our database, but not all. Wavelet decomposition preserves some temporal information at a range of time scales. The question of choice of wavelet type is usually based on expert knowledge. Factors such as the tradeoff between resolution in time and frequency domains must be considered. As with all such endeavours, the nature of the data and of the problem must be considered carefully before selecting a wavelet type.

However, Chapter 6 suggests that (some) flow series consist largely of peak events of
various sizes, which are to some degree parametrisable by a few template peak shapes. As we noted in Chapter 6 a characteristic shape is sometimes called a motif (for example see [81]), although the only shapes we are likely to encounter in natural flow series are the peak, and perhaps a double peak where events are closely spaced. Where artificial works such as dams and openable irrigation channels have been imposed, other structures might be seen. However, as we found in Section 6, streamflow peaks do occur on distinct time scales and with a range of different properties. A mix of different peak types will occur in most series, with a variable distribution.

Clustering those series with similar motifs together is an intuitive approach to the problem, and likely corresponds to a human method of classification. We can often pick short characteristic segments out of a long series by eye, and if two series have similar structures occurring frequently it seems sensible to say that the series have some similarity. Mutual information as a distance does not quite do this—rather than finding multiple series where motif A occurs, it finds series where a value A in one series corresponds in time to value B in another. The values A and B may or may not be one and the same, and may or may not occur as part of a recurring pattern.

The task of clustering by frequent patterns consists of two subproblems. Firstly, we need to identify frequently occurring patterns in each series (one or more appropriate motifs). The modified Apriori method of frequent pattern extraction discussed in Section 6.4 is an obvious choice of approach. The frequent subsequences discovered with this technique can be treated as the motif set, although the question of allowing for real-valued rather than categorical data must be examined. Then, a comparison must be made between the motifs of different series. As we are no longer considering direct correspondence in time, no limitations on corresponding time periods exist, and the full set of 36 longer flow series introduced in the introduction to this chapter can be used for investigation of this concept.

Motifs can be extracted with the modified temporal Apriori algorithm of Section 6.4, but before proceeding further we should apply our new and untested method to a standard dataset and compare the results to what is already known. Support shall be defined as in that section, but as we are not dealing with predictive rules on this occasion, no confidence threshold will be applied.

### 7.7.1 Verification of Method

The UCR Time Series Data Mining Archive contains many such standard datasets. The Reality Check dataset (detailed in [74]) is one of the smallest and most straightforward.
Figure 7.12, taken from [74], is a hierarchical clustering dendrogram for this data.

Figure 7.13 shows the corresponding hierarchical scheme generated with the motif method and the method used to obtain Figure 7.11. Data was discretised as discussed in Section 7.4.1. Support threshold (see Sections 2.5.2 and 6.4) was defined as 10% and maximum pattern length set at five.

A number of differences between the two structures exist. Most notably, the similarity of series 11 and 12 is not recognised by the motif clustering routine. Visual inspection of the two reveals the reason—series 11 is smooth and series 12 noisy. Thus, the series will be found to have different motifs. This sensitivity to noise could be easily removed by smoothing or resampling the series before clustering, but as we shall not use these techniques on the flow series we will not use them here either. The level of smoothing that is performed should in general be determined by the resolution and uncertainty of the data and other such concerns. Where the time sampling interval is sufficiently fine, additional points could also be considered in the gradient calculations—a process that would be similar in effect to smoothing the data in preprocessing.

Series 4 is most closely identified with 9 and 10, which does not seem unreasonable in a non-pointwise sense. The points of transition between state are much softer in series 1, which may be why it is not placed in this group high in the hierarchy. The rest of the clustering dendogram is visually sound, and at most points agrees with that of [74]. The counts matrix C (introduced in Section 7.3.1) is shown in Figure 7.14, which shows that discriminatory power of this method on the reality check data is considerable. Comparing this raster plot to the dendrograms of Figure 7.12 and 7.13, much the same information can be observed.

The number of frequent patterns (motifs) extracted from the reality check series varied from 11 to 41, with an average of 28. The original length of series was 1000 in all cases, with no missing points, but the nature of the motif extraction algorithm is such that the number of frequent patterns would not rise appreciably once a representative length of series had been processed. The modified Apriori algorithm is not as fast as its non-timeseries counterpart, but it is nonetheless very quick to run.

These results are very encouraging, and provide us with a solid indication that the method is sound.
Figure 7.12: Reality Check Hierarchical Clustering, as given by UCR Archive
CHAPTER 7. CLUSTERING TIME SERIES

Figure 7.13: Reality Check Hierarchical Clustering, Motif-based

Figure 7.14: C for Motif Distance between Reality Check Features
7.7. CLUSTERING WITH FREQUENT PATTERNS

7.7.2 Results from South-East Queensland Flow Series

With support threshold set at 10%, the full South-East Queensland flow database was processed and a set of frequent patterns obtained for each series. Considering the output from each dataset as a sparse vector of support values for frequent patterns with zero entries elsewhere, the Euclidean distance can be applied to generate a distance measure between frequency vector and hence flow series. Note that calculation of Euclidean distance between sparse vectors can be implemented efficiently, and does not require non-sparse expression of the two input vectors, as the pointwise comparison between two zeros does not contribute to the sum.

In Figure 7.15, clusters generated with the motif scheme are displayed. As only local time relations are considered, we are able to compare time series that do not overlap at any point. Note the presence of the strong group at middle right in both figures, which we have seen previously.

Examination of the counts matrix raster shows that stability of these clusters is quite high for most series, i.e. that strong tendencies exist for certain series to be assigned the same cluster. Although the resultant final clusters are very different to those generated with geographical distances, the trends are of the same magnitude for most series. This tells us that the motif-based method is capable of producing robust results.

Figure 7.15: Gauge Locations Coloured by Motif Cluster, Reduced and Full Datasets
7.8 Conclusion

In this chapter we have seen some of the ways that distance measures can be defined according to various purposes. Euclidean and L1 norm distances have been discussed, as well as mutual information and a new frequent motif distance. The first two of these consider pointwise differences, and the third pointwise correspondence between discretised series. The last quantitatively compares motif sets. All are useful tools for certain problems.

However, as noted in [75], a huge library of distance measures already exist, and for most tasks, one of these can be selected. A method that allows for fuzzy clustering would be a particularly interesting choice. Also noted in that presentation was the fact that simple Euclidean distance is hard to beat as a pointwise similarity measure. However, even when using only Euclidean distance we have seen that preprocessing options such as normalisation have a strong effect. Like all modelling processes, clustering can be done in a multitude of ways, and must be tailored to the problem at hand. We have shown how this can be done for a few different aims. Euclidean clustering matches up peaks and troughs in time, while an approach based on the spectra of frequently occurring patterns reveals similarity between the events that make up each series.

We have also illustrated problems with cluster stability and missing data that may arise, and suggested ways that these can be dealt with. Note that the cluster trend analysis and
missing distance estimation are not universal panaceas for instability and missing data, but intended to exemplify the issues that must be considered that will inevitably arise when real environmental data is used. Clusters generated in any given situation may not be stable, and so methods should be developed for investigating and representing it. Missing values are a common characteristic of environmental databases, and similarly they must be dealt with.

Avenues for further development of the techniques developed in this Chapter (and previously) exist, in particular the development of a more robust method of extracting clusters from the counts matrix $C$ and further exploration of the utility of the MDE algorithm. The motif-based clustering scheme of Section 7.7 will be addressed in more detail in Chapter 8. There, we apply the motif-based clustering method to a large, rich chemical trace dataset.
Chapter 8

Modelling Water Quality in the Plynlimon Area, UK

8.1 Introduction

In Chapter 6, the relationship between rainfall and electrical conductivity was investigated using data from a single catchment. Here, we examine an extensive and rich dataset from the Plynlimon catchments in Wales. The data are from not one but six catchments and 52 different measures of water quality in addition to streamflow. Most of these are chemical traces expressed in units of parts per million or parts per billion. Numerous studies have used data from this collection for diverse aims. See for example [108], [103], [106], [37], [46], [107], or [102].

Data was collected at six locations on the River Severn (known in Welsh as the Afon Hafren). Five of these are shown in Figure 8.1, taken from [101]. Two of gauges lie on the the river itself, two on the Afon Hore, a major tributary, and one on the Nant Tanllwyth, a smaller stream that feeds into the Hafren just below the lower Hafren measuring station. The Hore meets the Hafren below this point, close to the lower Hore recording point. South 2 Hore lies on a small stream feeding into the Hore between the upper and lower gauging stations. Mist water was also collected for analysis, although we shall not use this data here. The catchments of all streams are small, a few square kilometres or less in area (see [106] for details). Parts of the Plynlimon area are forested and periodically felled, and much of the analysis performed on this data by other researchers focusses on the effect of this process.

The upper Hafren gauge measures drainage primarily from moorland, and the lower
The Plynlimon Catchments mid Wales

Figure 8.1: Plynlimon Catchment Map, showing Gauge Sites

gauging region includes some forest, which has been felled in some areas over the period of record. This is similar to the distribution of vegetation in the Hore catchment, but [101] notes that there are differences in alkalinity between the Hore and Hafren waters, due to the presence of calcite veins in the Hore area. The Nant Tanllwyth catchment is entirely forested, as is the hillslope draining to the South 2 Hore measurement site. Flow is not measured at the South 2 Hore site, so flow from the low Hore was used in its place. [125] states that upper parts of the Hafren stream are steep and fast flowing, while the lower reaches are slower, reflecting a change in gradient.

Samples were collected approximately weekly for a period of years, although the exact sampling interval was not regular. Table 8.1 below summarises the record details of the six catchments. As usual, $N$ denotes the total number of instances. The area estimate for the South 2 Hore catchment is known to be uncertain, to around plus or minus 50% ([105]).

The water quality variables measured are Sulfate $SO_4$ (variable 2), Nitrate $NO_3$ (3), Chlorine $Cl$ (4), Calcium $Ca$ (5), Magnesium $Mg$ (6), Sodium $Na$ (7), Potassium $K$ (8), Silicon $Si$ (12), Dissolved Organic Carbon $DOC$ (13), Ammonia $NH_4$ (14), Phosphate $PO_4$ (15), and Fluorine $F$ (16) in parts per million (ppm), Ionised Hydrogen $H^+$ (9), Aluminium $Al$ (10), Boron $B$ (17), Barium $Ba$ (18), Beryllium $Be$ (19), Bromine $Br$ (20), Cadmium $Cd$ (21), Cerium $Ce$ (22), Cobalt $Co$ (23), Chromium $Cr$ (24), Caesium $Cs$ (25), Copper $Cu$ (26), Iron $Fe$ (27), Iodine $I$ (28), Lanthanum $La$ (29), Lithium $Li$ (30),
8.1. INTRODUCTION

<table>
<thead>
<tr>
<th>gauge number</th>
<th>Hafren</th>
<th>Hore</th>
<th>South Hore</th>
<th>Tanllwyth</th>
<th>Upper Hafren</th>
<th>Upper Hore</th>
</tr>
</thead>
<tbody>
<tr>
<td>area ((km^2))</td>
<td>8.7</td>
<td>3.2</td>
<td>0.35</td>
<td>0.90</td>
<td>3.6</td>
<td>1.6</td>
</tr>
<tr>
<td>start date</td>
<td>10/05/83</td>
<td>10/05/83</td>
<td>19/04/88</td>
<td>17/09/91</td>
<td>17/07/90</td>
<td>28/08/84</td>
</tr>
<tr>
<td>end date</td>
<td>20/02/01</td>
<td>20/02/01</td>
<td>20/02/01</td>
<td>20/02/01</td>
<td>20/02/01</td>
<td>20/02/01</td>
</tr>
<tr>
<td>N</td>
<td>964</td>
<td>975</td>
<td>638</td>
<td>454</td>
<td>540</td>
<td>864</td>
</tr>
</tbody>
</table>

Table 8.1: Plynlimon Dataset and Catchment Characteristics

Manganese \(Mn\) (31), Molybdenum \(Mo\) (32), Nickel \(Ni\) (33), Lead \(Pb\) (34), Praseodymium \(Pr\) (35), Rubidium \(Rb\) (36), Antimony \(Sb\) (37), Scandium \(Sc\) (38), Strontium \(Sr\) (39), Thorium \(Th\) (40), Uranium \(U\) (41), Yttrium \(Y\) (42), and Zinc \(Zn\) (43) in parts per billion \((ppb)\), Acid (44), AlkLow (45), and AlkHigh (46), \((pEql^{-1})\), acidity \(pH\) on the pH scale (11). Cation and CatSum (50), and anion levels AnSum (51) in \(\mu eq/L\), electrical conductivity Cond (47) in \(\mu S\), redox potential \(Eh\) (48) in \(\mu V\), ion balance IonBal (52) as a percentage, and water temperature Temp (49) in degrees centigrade. Flow (1) is given in cumecs \((m^3/sec)\).

AlkLow and AlkHigh are Gran alkalinity measures, taken with differing titration ranges. Differences between the two may reflect organic anion composition ([105]), but are unlikely to be significant. As we shall see in later clustering experiments, these two series are assigned the same cluster in all reported cases.

While the number of records in each catchment is not great, the number of variables is large. Although the record is not complete for some variables, the total number of measurements is still around 240000 values. Clearly, analysis of this data should be automated to the highest possible degree. Manual consideration of each variable individually is far too time-consuming, and interesting results would risk being lost in the huge pool of information extracted. Thus, this dataset is an ideal candidate for machine learning.

The problem differs from that described in Chapter 6 in several aspects. First and most obvious is the number of variables involved- preprocessing steps such as outlier identification, discretisation, and missing data treatment will all have to be automated if they are required. The discovery of the parametrisable nature of the peak and trough shapes in the last chapter was also made visually there, a luxury that will not be available to us here. However, it is unlikely that all 52 variables will be independent and represent different schemes of behaviour. Some will probably behave in a very similar fashion. Where this is true, we may submit for detailed analysis only one representative of the cluster.

For each location, series can be compared using pointwise methods such as Euclidean
distance. The records for each station correspond to distinct time periods, but it would also be very interesting to examine the likeness of and differences between, for example, potassium series for Hore and Upper Hore stations. Thus, a time-independent similarity measure is appropriate. Identifying these "like time series" is a challenge that will be addressed in Section 8.4.

Once the data has been preprocessed to an appropriate form, we shall proceed using the motif-based techniques of Chapter 6 and Section 7.7 to investigate the properties of the extensive and rich Plynlimon dataset. The motif-based techniques suggest themselves because we would like to investigate which series respond in similar ways to whatever drivers exist. As we shall see later, certain series are characterised by (for example) double peaks. Also, we do not have to limit our analysis to corresponding time periods between locations. For the sake of brevity the analysis process will be discussed in detail only for the Hafren station, although results will be given for all catchments.

In this way we hope to satisfy two distinct aims: to investigate and discover new knowledge from the Plynlimon data, and to explore the capability and utility of the motif extraction method developed in previous chapters. In Section 8.2, we examine the data in detail, considering missing values, sampling resolution, and the presence of outliers. Extracted characteristic motifs are considered in Section 8.3, and the Hafren series are clustered by Euclidean and motif means in Section 8.4. Variables are considered across locations in Section 8.5, and general conclusions drawn in Section 8.6.

8.2 Data Quality and Preprocessing

It is known (from [104]) that some of the Plynlimon data is of low quality. Certain variables are known from comparative measurement studies to be poorly captured, and significant gaps exist in most records. Outliers certainly exist and we shall encounter some of these.

The first task is to investigate the quality of the record. Considering the number of missing points in each variable, we find that 32 out of the 52 variables contain less than 1% missing points. These 32 contain only a handful of missing points. A further 5 variables fall into the 1%-5% interval, but 10 variables have between 40% and 60% of measurements missing. Examination of these reveals that the missing points are not scattered throughout the series, but take the form of large missing blocks usually at the beginning or end of the time period in question. Outside these regions, the data is sampled at the same intervals as are the other variables. Thus, some usable periods exist.
8.3. CHARACTERISTIC MOTIFS WITHIN THE HAFREN DATASET

Sampling intervals do not remain constant throughout the record. Weekly sampling appears to be the norm across all variables, with more than 80% of intervals weekly or near-weekly. In a few places, high-resolution measurements were taken, and in a very few cases longer sampling intervals exist. The weekly interval is sufficiently large to remove the need for the smoothing which was required in Chapter 6. In fact, it is so large that concern may be raised regarding proper capturing of system dynamics, as we saw in Chapter 5. Fortunately the periods of finer resolution measurement allow us to test this.

While not all variables will respond at the same time scale, it is reasonable to assume that they are all (to some degree) driven by flow. Therefore if the behaviour of streamflow is not well represented, the amount of information we can extract from the dataset will be limited. Figure 8.2 shows a portion of the raw and resampled Hafren flow record for 1988. The resampled series was forced to a weekly interval by interpolation, removing points from regions with higher resolution. Where long (> 2 week) gaps in time exist, missing point markers were inserted rather than approximate values calculated. On the left we see that the addition of an extra data point brings very little new information, but on the right, the resampled series has missed an entire peak. This is a significant issue, but the size of the peak in question is only about one tenth of the average peak height. Thus, its absence from the record is unlikely to cause significant problems. However, most peaks are represented by a single extreme point, so it is possible that larger events have been missed.

Sparse sampling also leads to problems when identifying outliers. For example, the flow series contains one point an order of magnitude greater than any other. Unfortunately, no measurements were taken for one week before and after the point in question to confirm or deny the validity of the extreme value. A number of other variables also exhibit from one to a handful of points that may or may not be outliers. Aside from expert knowledge of the system, the only existing avenue for investigation of outliers lies in the behaviour of other variables at this time. When relationships between variables have been established, this question will be addressed again.

8.3 Characteristic Motifs within the Hafren Dataset

Recall the discretisation scheme introduced in Chapter 6 and also used in Chapter 7.

0 0 flat

1 U rising
Using these categories, the Hafren dataset (gauge 54022) was processed into discrete form, and frequent motifs extracted by the modified Apriori algorithm of Section 6.4. As always, we shall use a support threshold to determine frequency, and place no other conditions on extracted motifs. Before beginning the clustering and comparison process, it is instructive to consider the nature of the patterns that occur frequently in the different variables of this dataset.

First, consider the flow series. We would expect peak shapes, as were extracted from the South-East Queensland and Yarrangobilly streamflow data in Chapters 6 and 7. Here we shall allow consideration of patterns with internal zeros, where before it was assumed that a flat shape represented the end of the current pattern. However, leading zeros are still disallowed. Applying a 10% support threshold and a maximum pattern length of 5, we find the following motifs:

2 D falling
3 U+ rising, following a lesser rise
4 U- rising, following a greater rise
5 D+ falling, following a lesser fall
6 D- falling, following a greater fall
These are listed from left to right and top to bottom in order of number of appearances. Most notably, the secondary categories of \( U^+ \), \( U^- \), \( D^+ \), \( D^- \) never occur in the frequent set. The motifs for the Hafren flow series are all simple peaks shapes or parts of peaks. The double peak \([U \ D \ U \ D]\) is frequent, and its subpattern \([D\ U]\). Examination of support values tells us that \([D\ U]\) also exists outside of \([U\ D\ U\ D]\) sequences. The \([D\ 0...]\) pattern also recurs often, which tells us that periods of activity most frequently end with a drop.

Next consider the Hafren pH series. It produces a completely different motif set, listed in order as above. Here we see dominance of rising and falling sequences with length up to three, with very variable convexity properties. These rarely end with a zero.

The first 500 points of the original flow and pH series for the Hafren measuring station are shown in Figure 8.3. Both series are normalised to the scale 0-1 in that figure, and the very different modes of behaviour can be clearly observed. Missing points are marked by negative numbers. Note also that the seasonal variation in pH is not picked up by the motif characterisation.

At this point we should consider the effect of our choice of parameters. As we are not filtering frequent patterns by any means other than support (for example confidence, which is discussed in Section 2.5.1), this can be investigated easily by varying that threshold. With a minimum support \( s \) of 10%, the mean number of motifs discovered in the Hafren
variables was 23. Thus, the representation of a Hafren series contains only 23 real numbers
and 23 integers, each pair being a motif support value and the identifier of the motif itself.
It is thus a very compact representation. Table 8.2 summarises the variation in number
of motifs for the Hafren dataset.

In two of the water quality variables, \textit{CatSum} and \textit{AnSum}, only the zero marker is
identified as frequent. These series are indeed very flat, each with one massive outlier.
The outlying event only appears at one point in time (the points on either side are not of
unusually high magnitude), but does occur in both variables. A very large flow magnitude
was also recorded on this day. Therefore this outlier is likely to represent a real-world
extreme event.
Recall that the 0 marker is given to differences of magnitude less than 10% of the average change. Clearly, the magnitude of change in this one value causes this threshold to be set too high for all other differences to be assigned anything but 0. A more sophisticated 0 threshold could be defined to ameliorate this problem, or such extreme values removed in preprocessing. However, the extreme value maybe the most important point in the CatSum and AnSum series, and it is interesting to note that the algorithm has set them apart (albeit for undesirable reasons). As the outlying point cannot in all fairness be removed from the dataset, we shall leave the results as they are, and observe the effect this has on later clustering and comparison between series.

Returning to the flow and pH series, when \( s = 20\% \), four and 13 patterns are identified respectively. Reducing the number of patterns in the list of flow motifs to only the first four would destroy a great deal of interesting information and leave us with very little. On the other hand, when \( s = 5\% \), the maximum number of patterns extracted is 51. There are 25 elements in the list of frequent (\( s = 10\% \)) pH sequences, and already there is a great deal of information presented to the user. The threshold of 10% seems therefore to be reasonable, although we shall continue to experiment with the 5% and 20% parameter values.

We could undertake a detailed examination of any variable of interest at this point, but there are too many features in this dataset to examine each one separately, so we will proceed to the clustering stage and divide the Hafren variables into manageable groups.

### 8.4 Clustering the Hafren Time Series

In this section we shall cluster the Hafren dataset with the k-means algorithm as applied in Chapter 7. First, in Section 8.4.1, we shall use the motif distance developed in the previous chapter, considering the effect of outliers. Then, in Section 8.4.2, clustering with the Euclidean distance will be performed and the results compared with those obtained by motif distance clustering.

#### 8.4.1 Clustering the Hafren Time Series with Motif Distance

From the previous section, we know that two outliers exist, in the CatSum and AnSum series (50 and 51), and that this has resulted in poor motif representation in these variables. Therefore, let us test the effect this has on the clusters resulting from the Hafren dataset by extracting two counts matrices. The first shall be calculated with the full set of variables,
and the second with $CatSum$ and $AnSum$ removed. Figure 8.4 shows both. In all respects, these counts matrices $C$ as calculated as in Chapter 7.

Clearly, removal of the anomalous variables from the dataset has little effect on the counts matrix, as we can see very little difference between the two rasters in Figure 8.4. Hence we may say that it rarely changes the cluster assignments of other variables. Also, it can be observed that a strong correspondence between $CatSum$, $AnSum$, and series 15, Manganese, (and to a lesser degree series 32 and 1, Molybdenum and Flow). All of these series have a high proportion of zero values. Molybdenum in particular is also known to be a poor quality record that does not appropriately represent system dynamics ([104]), and this can be observed directly by plotting against time. Negative values have been recorded, and measurement resolution appears very low. However, for all series except 50 and 51, several motifs of length greater than one have been identified. To test the stability of the method with respect to changes in the threshold support $s$, count matrices generated with $s = 5\%$ and $s = 20\%$ are given in Figure 8.5. This figure shows that $C$ is remarkably stable.

Let us examine in detail two strong clusters and their constituents. Using the $s > 10\%$ motif set and defining a count threshold $Cthr = 70\%$, 12 clusters are identified from the counts matrix shown on the left hand side of Figure 8.5. These are listed below. Note that series 45 and 46, $AlkLow$ and $AlkHigh$, are grouped together.

1 5, 11, 12, 45, 46
2 8, 22, 27, 43
3 25, 28, 40
Choosing the first cluster arbitrarily and visualising the members, we arrive at Figure 8.6. The data have been linearly normalised to lie between zero and unity, but two of the series contain extreme events (possible outliers), so they appear lesser in magnitude. All series show a strong correspondence in time, which is not used as a criteria in the motif distance measure at all. Also note the presence of [U U- D] peaks in all variables.

A time series plot of 12 series is crowded and difficult to interpret, so $Cthr$ was relaxed to 50% so the number of clusters reduced to five. A segment from each of the first series from each cluster (variables 1, 2, 14, 8, and 4) is plotted in Figure 8.7, linearly normalised. The different characteristics of each series can be observed. For example, the Chlorine
series (in pink) appears to have a characteristic double peak, while the Nitrate series (in red) is flat for long periods with a few sharp peaks. The cluster 1 variables, as illustrated in Figure 8.6, hold a distinct trend towards motifs that rise sharply and then more gently to a peak. Double peaks also seem very common - the Chlorine series is also a member of this cluster.

8.4.2 Clustering the Hafren Time Series with Euclidean Distance

It has been stated (in [75] for example) that it is difficult to beat the Euclidean distance as a times series similarity measure, and thus it is a standard benchmark. Therefore, clustering under Euclidean distance was performed on the Hafren data, linearly normalised to lie between zero and unity. There is no time period where measurements exist for all variables, but the only two variables that do not have common points are 44 and 46, *Acid* and *AlkHigh*. All other distances were calculated on a subset of the dataset (of length 216 to 253 points). The Missing Distance Estimation algorithm of Section 7.3.2 was used to estimate the distance between *Acid* and *AlkHigh*. The time periods selected for normalisation and distance calculation were subject to cursory visual inspection to check for the presence of outliers, so extreme values could not adversely effect the normalisation process.

The resulting clusters (as at $C_{thr} = 50\%$) are:
8.4. CLUSTERING THE HAFREN TIME SERIES

Figure 8.7: Hafren Cluster Representative Variables, Motif Distance

1 5, 6, 11, 12, 30, 39, 45, 46, 49, 52
2 1, 9, 14, 15, 17, 18, 22, 24, 26, 29, 34, 37, 40, 41, 48
3 2, 3, 4, 7, 16, 19, 20, 23, 25, 31, 32, 44, 47, 50, 51
4 8, 13, 21, 27, 28, 35, 38, 43
5 10, 33, 36, 42

These are considerably larger than the corresponding set of clusters generated with
the motif sets, and fewer in number. Inspection of the counts matrix for the Euclidean
case shows that inter-variable relationships are somewhat stronger than shown in the motif
counts matrix, as we would expect.

Note that the Euclidean cluster 1 contains many of the same elements as the motif
cluster 1. Figures 8.8 and 8.9 are views of the series in the Euclidean cluster 1 at different
scales. The resolution on the x axis is weekly, and will remain so throughout this chapter,
but Figure 8.8 covers roughly a range of 35 weeks and Figure 8.9 just under five years.

In Figure 8.8, it is clear that series only behave in a similar fashion in some regions,
but for much of the time they are acting very differently. However, Figure 8.9 (although
confused by the large number of series) demonstrates that the general trend is similar for
all elements of the cluster. Thus it is clear that the Euclidean and motif distances are
measuring very different aspects of series similarity, as they were designed to, although there is some correspondence between Euclidean and motif clusters.

### 8.5 Comparison of Variables Across Locations

The motif-based similarity measure does not rely on pointwise comparison in time, so it is possible to compare series that do not correspond in time. Although a common time period does exist between all six measurement locations, we shall not use pointwise measures to produce similarity information to compare with that obtained from the motif sets. We have already shown that the two approaches consider very different aspects of series similarity, and therefore, such a comparison does not provide a measure of accuracy. Instead, we shall look at what knowledge can be extracted from similarity information across different locations.

Choosing the variables shown in Figures 8.6, distance matrices were generated for each, across the six locations. All distance values were linearly normalised to lie between zero
8.5. **COMPARISON OF VARIABLES ACROSS LOCATIONS**

![Hafren Euclidean Cluster 1 Variables, View 2](image)

Figure 8.9: Hafren Euclidean Cluster 1 Variables, View 2
and unity for easy visual comparison. Figure 8.10 shows distance matrices for motif cluster 1 variables as raster plots. Measurement locations are ordered along the x and y axes in the following order: Hafren, Hore, Upper Hafren, Upper Hore, South 2 Hore, Tanllwyth.

From this figure, it can be stated that the motif structure does not necessarily remain constant between locations. It may be intuitively reasonable that structures would be preserved between the two Hafren gauges and also between the upper and lower Hore gauges, but this does not appear to be the case. However, it was noted in the introduction that the time scale of flow response varies between the upper and lower reaches of the Hafren, which suggests that a different motif structure will emerge from each. The vegetation cover is also similar between Upper (lower) Hore and Upper (lower) Hafren, rather than Hore and Upper Hore or Hafren and Upper Hafren.

In Figure 8.10, distances are low (towards the blue end of the spectrum) for Hore and Upper Hore (locations 2 and 4) in the first, second, and fourth (Mg, pH, and AlkLow) rasters. Similarly, distances are low for the Hafren and Hore (locations 1 and 2) in pH and Si distance matrices. Therefore we can be more specific than crudely choosing statements such as 'Upper Hore and Hore chemical behaviours are similar' or 'Hore and Hafren chemical behaviours are similar', even within a cluster. The scope for extraction of this kind of information is enormous.

To quantify the typical distance between unrelated variables, let us consider the distance
matrices between variables of different cluster and location. Generating distance matrices for two sets of five randomly selected variables from randomly selected catchments and returned an average distance of 0.26, where the average distance between variables in Hafren cluster 1 was only 0.11.

8.6 Conclusions on the Utility of Motif Distance and Series Distance Measures in General

As we have seen in this chapter and the previous one, series similarity measures are useful analytical tools for clustering and for more direct investigation. Information on the correspondence of behaviours across different locations can be extracted. Additionally, the analysis task can be reduced by clustering like series. Pointwise and compression based similarity measures produce very different results, because they use vastly different concepts of what makes two series similar. The nature of the data and also the goals of the analysis task must be taken into account when choosing one measure over another.

The motif-based distance introduced in Chapter 7 and applied and extended here shows strong promise as a method for similarity quantification. We have shown how it can be used to investigate a large real-world dataset, and noted a number of interesting features. Even before any distance measure is considered, the motif extraction method returns useful information on the composition of input series. While this knowledge could be obtained visually, that approach is not only highly labour intensive but also subjective, while our algorithm is quantitative, repeatable, and objective.

There are many outliers in the Plynlimon dataset, but the extraction of characteristic motifs is not sensitive to outliers because a single point does not influence the shape of many candidate patterns. However, a single very extreme event can flood the shape construction algorithm as it now stands, by setting the threshold for flat segments too high.

Generated clusters have stability similar to those generated with Euclidean distance, and we have seen that the method identifies visually reasonable clusters. The effect of varying the minimum support defining a motif does not appear to be strong, at least for this data where several very different schemes of behaviour can be identified.
8.7 Further Work

It must be acknowledged that only one specific implementation of the motif extraction idea has been implemented here, although we have obtained good results with it. Options to be explored include additional discretisation schemes (for example, removal of convexity information, or use of numerical categories, or simply alteration of the flatness threshold function). The time scale could also be visited in greater depth, as a process such as streamflow generation may occur at different scales in different locations.

Regarding scale of motifs, note that (by the Apriori property, see Section 2.5.2) if a motif is not frequent, no additional information will make it so. Therefore, adding some reference to event magnitude into the pattern representation will not result in new frequent patterns, although some may be lost. If a scale-free pattern is not frequent, the scaled representation will not be frequent either. However, some valuable information is probably being lost by discretising with the scheme we have chosen. Possible avenues for the inclusion of this information include additional categories based on magnitude, flagging of unusually large positive and negative gradients, or construction of distributions of motifs based on magnitude. The investigation of electrical conductivity motif magnitudes in Section 6.6 suggests that the last of these options may be very fruitful.

Rather than further comparison with a pointwise distance metric, a wavelet spectral approach could be used as a new baseline, as this technique looks at similarity between the shape of time series at various scales. It would also be valuable to compare our results with those obtained by more recent developments such as those discussed in [80].

The next step would be to attempt to answer specific questions regarding the Plynlimon dataset, and to seek physical interpretation of the clusters. One possible avenue of enquiry would be the changes in stream chemistry that occur when a forested catchment is logged. These changes could be compactly quantified for a particular variable by considering the motif set extractable from periods before and after the event. Other questions should be decided upon in close collaboration with researchers working in the catchment on real-world problems. An extended exploration would give further indication of the capability of the method, and also perhaps discover new and novel knowledge about the system.
Chapter 9

Conclusions

9.1 Introduction

In this chapter, we briefly revisit the projects undertaken as part of the thesis, and summarise both the modelling achievements and lessons learned in the process. Conclusions can be drawn from two perspectives - throughout the case studies, we seek to extract system knowledge, and also to consider the tools required for this operation. As noted in Chapter 1, only well-established techniques are applied in Chapters 3, 4, and 5, but in Chapters 6 to 8, new techniques are introduced and explored. Thus, we comment on methodological advances as well as the information generated by the application of these and other data mining algorithms.

9.2 Review of Data Mining Techniques

This section was intended both as a reference for further chapters and as a rough guide and introduction to the field of data mining for environmental scientists. The central disciplines of data mining such as clustering, classification, and rule extraction were discussed with examples of how they can be applied. Outlines of a few specific algorithms were given, to provide some insight into the design and function of data mining algorithms.

Preprocessing techniques were also discussed, as it is our belief that these are an integral part of the knowledge extraction task. Later application chapters demonstrate that this is very much the case. Visualisation, uncertainty quantification, outlier identification and treatment, and model performance reporting were included along with a variety of other topics, many of which were subsequently used in Chapters 3 to 8.
9.3 Gully Erosion Location Classification

The gully erosion project was the simplest and most straightforward of our analyses. Once the problematic asymptotic behaviour of the gully density measure was controlled, C4.5 and ADtree decision tree extraction algorithms produced clear results. We were able to identify the strongest determinants of gully erosion in the Ben Chifley catchment (soil class, elevation), and poor quality with the slope variable was exposed through surrogacy. We also noted that a division between zero-low erosion and medium-high erosion may be more natural than erosion present/absent.

Visualisation proved an invaluable tool in both the preprocessing stage and for presentation of results. In particular, distribution plots such as the one illustrating the effect of soil type (Figure 3.3) on severity of gully erosion, and the increasing trend towards gully erosion with rising elevation in Figure 3.4 revealed and succinctly presented useful information.

9.4 Rainfall Intensity Extraction

In this chapter we first encountered the issue of problem definition. While the goal, discovering some information about sub-daily rainfall intensity, was known, rainfall intensity itself remained undefined. Unlike the gully erosion data, which was given as a complete dataset and stemmed from the investigation [136], we followed this problem from data collection to the point of application in rainfall-runoff models. Climate data was collected from various sources, and the definition of rainfall intensity developed and defined into classifier variables using a flow per interval threshold and a more sophisticated distribution clustering approach. These were then investigated with a range of different classification algorithms, and under a variety of parameter, resampling, and feature selection choices.

The best classification schemes correctly identified about half the Adelaide and two thirds of the Brisbane intense events, with negligible incorrect assignment of non-intense days. Thus, the method is well placed to provide valuable information to daily rainfall-runoff models.

Like the gully erosion project, the only algorithms used here were tried and tested examples implemented in pre-existing data mining software suites. It serves to demonstrate the utility of such packages, and the type of potentially important environmental problem that can be addressed with data mining.

We found that, particularly in the Brisbane dataset, strong characteristic events exist,
such as Brisbane Summer storms characterised by high temperature, high humidity, a
Summer date, and mid to high daily rainfall. Intense rainfall events in Adelaide occurred
more evenly throughout the year, but many were still identifiable by examination of daily
variables such as humidity, wind speed, presence of lightening, and daily aggregate rainfall.
It is also notable that for both regions the number of non-intense events incorrectly labelled
by the models as intense was very low.

9.5 Jugiong, Muttama and Tarcutta Water Quality

Here, substandard data quality was a serious problem. Models describing the relation­
ship between streamflow and water quality were required for a larger water quality project
([135]). Uncertainty of suspended sediment concentration data coupled with long sam­
pling intervals rendered analysis with classical methods fruitless, and we were only able to
make small gains. By adding new variables describing time of year and local flow profile,
predictive capacity was increased, although only modestly.

The value of this project lay in the demonstration that data sparsity is a very real
problem, which can in some cases be ameliorated by the introduction of new variables.
While data mining can produce good results for many problems, and additional methods
can be applied to increase the amount of information generated, the possibilities are still
limited by available data. Where uncertainty exists and cannot be removed, it and its
limiting effect on analysis must be acknowledged.

Local flow profile as a determinant of suspended sediment concentration supports the
conclusions of (for example) [91], and suggests that a modification of sampling procedures
for both flow and water quality to take this into account may yield useful results and
allow for the construction of a much more useful water quality model. This was found to
be the case in Chapter 6, where a water quality modelling scheme incorporating higher-
resolution information was developed. This approach then led to the motif-based series
characterisation scheme explored and applied in Chapters 7 and 8.

9.6 Electrical Conductivity Modelling

A high quality electrical conductivity and flow dataset was obtained to further develop the
concepts arising from the previous chapter. This data had 15 minute resolution in both
variables, and a record length of several years. A small number of outliers were treated
manually, and as adjacent values were highly coupled, the series were reduced by piecewise linear decomposition. The resulting series were shown to fit the original data well, with little loss of information.

Frequently occurring patterns were then identified from scale-free discretised representations of electrical conductivity and flow series using a modified Apriori algorithm, and these were found to be intuitively appropriate peaks in streamflow and troughs in conductivity. When scale was added back into the analysis, it was found that while neither the set of peaks nor the set of troughs consisted of congruent shapes, the shapes of both groups were parametrisable by a magnitude value. In other words, the trough depth or peak height determines near-uniquely the shape of the rest of the pattern. However, relations regarding the time scale of the characteristic patterns were less clear.

Considering two-dimensional relationships, it was found that although flow peaks and conductivity troughs tend to occur close to one another in time, electrical conductivity events can begin before the corresponding flow peaks. Timing errors in the data were unlikely, and the measurements were taken at the same location. This point proved to be of great interest to the scientists who collected the data.

A model relating peak magnitude to trough depth was established, and it was found that the initial value of the electrical conductivity had a very strong influence on the minimum value of the conductivity. Again, this information has now been passed to the investigating hydrologists for further consideration. The project was highly successful both as the development of a tool and in the information generated. The event-based representations used by the method mirror intuitive understanding of the input time series, while also providing quantitative information.

9.7 Flow Time Series Clustering

In this chapter we returned to using tried and tested methods, although it was found very quickly that new approaches were required to account for missing data and instability. Series distance quantification and clustering of flow series is an important task for regionalisation studies, where a parameter set calibrated on one catchment is extended to cover another, in which there is not sufficient data to calibrate the model. In this way, predictions for ungauged basins can be obtained. To do this successfully and with confidence, it is important to understand and measure the likeness of flow series from different locations and with different characteristics. Considering a database of streamflow records from
a variety of streams and rivers in South-east Queensland, Australia, and the problem of clustering, we found that the question of problem definition was all-important.

A purely geographical clustering was performed using gauge location data, but in many cases we would like to know which series are similar rather than which measuring stations are close together. Nearby stations are more likely to be similar than distant stations, but they may also have very different characteristics. Similarity can be considered as a pointwise Euclidean metric between normalised or unnormalised series (it may be necessary to differentiate between those water courses with baseflow and those without), as a measure describing where the gradients of the series have the same sign, or based on a compression method where reduced representations are compared rather than entire series. Several different definitions of distance, and their possible uses, are demonstrated and explored.

The gradient-based mutual information method used a discretisation scheme similar to that deployed in Chapters 5 and 6, where it was shown to be informative. The differences between clusters generated with Euclidean distance and this measure were explored briefly, and it was seen that the two approaches generate useful but different perspectives.

No common time period existed for all series in the dataset. Therefore, an algorithm (Missing Distance Estimation or MDE) was developed to estimate those distances that could not be directly obtained. Performance analysis of this technique by hiding some distances, and comparing with the estimates so generated, revealed it to be very sound where missing values are reasonably sparse.

The problem of generating stable clusters was also addressed. Rather than moving to a more complex clustering algorithm when the k-means approach failed to generate absolutely stable clusters (under any distance measure), an aggregation of multiple runs was used to produce cluster trends and also to explicitly indicate stability.

Lastly, a series characterisation method based on characteristic patterns or motifs was developed using the shape extraction method of Chapter 6. It had the advantage that no common time period was required. Because this was untested and we could expect results significantly different from those generated by pointwise distance measures, a testing dataset with known behaviour was obtained. A hierarchical clustering was performed with our new methods and compared with the reference Euclidean distance hierarchical clustering provided with the data. The method was shown to produce good results, so it was then applied to the flow dataset.

In this way, uses of different distance measures were explored, as well as the problems that may arise when using imperfect data and algorithms. We demonstrated ways to deal
with these, as well as the benchmarking of a new method.

9.8 Plynlimon Series Investigation

The motif-based series characterisation used in Chapter 7 was applied here to a real world dataset to explore the utility of the concept: and see what information could be extracted from the extensive and very rich Plynlimon in-stream chemistry data. We showed that individual series can be characterised by a small number of motifs (less than 10 in many cases), and generated clusters using this method and the reference normalised Euclidean distance. The value of the motif representation, which considers the similarity of the set of characteristic events, became clear for both individual series characterisation and clustering.

Series of different variables were compared for each of the six Plynlimon locations, and certain variables were selected for comparison across locations. It was found that inter-location distances between a particular variable were in general far less than the average distance between variables, but significant variation does occur.

In further work, the motif characterisation could be extended to include a time-warping component, which would be particularly interesting in inter-location comparisons. The richness of the Plynlimon dataset also suggests that specific questions could be answered with this method.

9.9 General Conclusions

We have applied well-known data mining techniques to a number of environmental problems, including gully erosion location identification, intense rainfall event characterisation, and flow series clustering, as well as adapting existing methods to our own ends for water quality, electrical conductivity modelling, and series clustering. New methods were developed for series characterisation and clustering by frequently occurring patterns. Insight into the properties of datasets describing the aforementioned environmental systems has also been gained, along with understanding into the physical processes involved.

In the course of addressing these problems, numerous issues arose from poor data quality, missing values, and instability of output. We dealt with these where possible, and rigorously reported where further progress was not possible. Preprocessing, visualisation, treatment of uncertainty, and presentation of results were included as significant steps
in the knowledge extraction process, greatly increasing the utility of more specific data mining algorithms and improving the quality of results. All cases studies demonstrated good practice data mining, and the value of rigorous application of such principles.

Clustering, classification, rule extraction, and preprocessing techniques were introduced, and all of these were used at least once, and modified to meet the needs of environmental problems. In this way, we hope to have provided a demonstration of the ways in which data mining can be used in the environmental sciences.
CHAPTER 9. CONCLUSIONS
Bibliography


BIBLIOGRAPHY


[76] EJ Keogh and MJ Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In Proceedings of


