School of Computing

# Remote Sensing, Geographic Information Systems (GIS) and Bayesian Knowledge-Based Methods for Monitoring Land Condition

Peter Albert Caccetta

This thesis is presented as part of the requirements for the award of the Degree of Doctor of Philosophy of Curtin University of Technology

March 1997

## Certification

I certify that the work presented in this thesis is my own and that all references are duly acknowedged. This thesis has not been submitted previously, in whole or in part, in respect of any academic award at Curtin University of Technology or elsewhere.

Peter A. Caccetta

March 1997

## Acknowledgements

I would like to thank the Curtin University School of Computing, Digital Equipment Corporation and the CSIRO Division of Mathematics and Statistics for providing the funding which made this research possible. I would also like to thank the farmers/farming groups and the various government agencies, including the Department of Land Administration and Agriculture Western Australia, for providing the data used in my research.

In the process of doing my research, I am indebted to those that inspired and encouraged me in the right direction. These people include Geoff West, Mark Gahegan, Norm Campbell and Harri Kiiveri.

Finally, I would like to thank all the people that I worked with on the various projects undertaken in the course of my work.

## Abstract

This thesis considers various aspects of the use of remote sensing, geographical information systems and Bayesian knowedge-based expert system technologies for broad-scale monitoring of land condition in the Western Australian wheat belt.

The use of remote sensing technologies for land condition monitoring in Western Australia had previously been established by other researchers, although significant limitations in the accuracy of the results remain. From a monitoring perspective, this thesis considers approaches for improving the accuracy of land condition monitoring by incorporating other data into the interpretation process.

Digital elevation data provide one potentially useful source of information. The use of digital elevation data are extensively considered here. In particular, various methods for deriving variables relating to landform from digital elevation data and remotely sensed data are reviewed and new techniques derived.

Given that data from a number of sources may need to be combined in order to produce accurate interpretations of land use/condition, methods for combining data are reviewed. Of the many different approaches available, a Bayesian approach is adopted.

The approach adopted is based on relatively new developments in probabilistic expert systems. This thesis demonstrates how these new developments provide a unified framework for uniting traditional classification methods and methods for integrating information from other spatial data sets, including data derived from digital elevation models, remotely sensed imagery and human experts.

Two applications of the techniques are primarily considered. Firstly, the techniques are applied to the task of salinity mapping/monitoring and compared to existing techniques. Large improvements are apparent. Secondly, the techniques are applied to salinity prediction, an application not previously considered by other researchers in this domain. The results are encouraging. Finally limitations of the approach are discussed.

# Contents


**1 Introduction** **1**

**2 Literature Survey** **10**

2.1 Introduction . . . . . 10

2.2 Classification . . . . . 12

2.2.1 Background . . . . . 12

2.2.2 Processing of Remotely Sensed Data . . . . . 13

2.2.3 Classification using Remotely Sensed and Other Data . . . . 15

2.3 Reasoning With Uncertainty In Knowledge-Based Systems . . . . . 17

2.3.1 Background on Bayesian Probability Theory . . . . . 19

2.3.2 Basic Concepts of Dempster–Shafer Theory of Evidence . . . 24

2.3.3 Exploiting Domain Structure — Propagating Beliefs in Hypertrees . . . . . 33

2.3.4 Bayesian Networks — A Graphical Tool . . . . . 44

2.3.5 Bayesian Learning Strategies . . . . . 47

2.4 Summary . . . . . 49

2.5 Deriving Landform Features . . . . . 50

2.5.1 Background . . . . . 50

2.5.2 Potential Applications for Digital Elevation Models in Agriculture . . . . . 53

2.5.3 Processing Techniques . . . . . 55

2.5.4 Landform Feature Extraction . . . . . 55

2.5.5 Water Accumulation Algorithms . . . . . 64
2.5.6 Drainage Networks . . . . . 67
2.5.7 Summary . . . . . 73
2.6 Summary . . . . . 74

**3 Problem Domain** **76**
3.1 Introduction . . . . . 76
3.2 The Moora / Kalannie-Goodlands Area . . . . . 79
3.2.1 Background . . . . . 79
3.2.2 The Moora / Kalannie-Goodlands Database . . . . . 81
3.3 The Upper Kent River Catchment . . . . . 87
3.3.1 The Upper Kent River Catchment Database . . . . . 90
3.4 Summary . . . . . 90

**4 Implementation of algorithms relating to Bayesian networks** **92**
4.1 Introduction . . . . . 92
4.2 Algorithm Implementation . . . . . 94
4.3 Optimal Triangulation . . . . . 97
4.3.1 Terminology . . . . . 98
4.3.2 The Algorithm . . . . . 100
4.3.3 Results . . . . . 102
4.4 Processing Data that Contain No Attribute Uncertainty . . . . . 104
4.5 The EM Algorithm and its Implementation . . . . . 107
4.6 Summary . . . . . 110

**5 Experiments in Landform-Morphology Feature Extraction** **113**
5.1 Introduction . . . . . 113
5.2 The Digital Elevation Model . . . . . 115
5.2.1 DEM Construction . . . . . 115
5.2.2 Errors in the broad-scale DEM . . . . . 120
5.3 Some Initial Partitioning Experiments . . . . . 123

5.3.1 Application of Existing Techniques . . . . . 123
5.3.2 Height-Constrained Valley Extrapolation . . . . . 125
5.4 Flat-Area Classification . . . . . 136
5.4.1 Existing Approach . . . . . 136
5.4.2 New Approach . . . . . 138
5.5 New Approach — Combined Accumulation Algorithm . . . . . 146
5.6 Summary . . . . . 152

**6 Experiments in Combining Data — Moora/Kalannie-Goodlands 157**
6.1 Introduction . . . . . 157
6.2 The Xantippe Catchment . . . . . 159
6.3 Simple Models for Integrating Remotely Sensed Data With Ground Data . . . . . 160
6.3.1 Data . . . . . 161
6.3.2 Existing Models . . . . . 163
6.3.3 New Models . . . . . 167
6.3.4 Comparison of results . . . . . 172
6.3.5 Conclusions . . . . . 182
6.4 Simple Temporal and Spatial Models . . . . . 184
6.4.1 Temporal Models – MODEL3 and MODEL4 . . . . . 185
6.4.2 Spatial Models . . . . . 187
6.4.3 Classifier Comparisons . . . . . 192
6.5 A Knowledge-Based Approach to Combining Multiple-Sources of Data . . . . . 204
6.5.1 Knowledge Elicitation . . . . . 204
6.5.2 Data Sets . . . . . 205
6.5.3 Temporal Aspects of Land-Use . . . . . 209
6.5.4 Network Configuration . . . . . 211
6.5.5 Parameter Determination . . . . . 214
6.5.6 Application of the EM Algorithm – Training the Network . . 214

6.5.7 Results . . . 216
6.5.8 Conclusions . . . 220
6.6 Summary . . . 221

**7 Application — Predicting Salinity** **227**
7.1 Introduction . . . 227
7.2 Knowledge Elicitation . . . 229
7.3 Extracting Landform Features and Related Variables . . . 231
7.4 Proposed Model . . . 235
7.5 Estimating the Network Parameters . . . 240
7.6 Results . . . 241
7.6.1 Computational Aspects . . . 241
7.6.2 Comparison with Training Data . . . 242
7.6.3 Comparison with Validation Data . . . 247
7.6.4 Catchment Summary . . . 249
7.7 Summary . . . 254

**8 Further Research** **257**
8.1 Model Sensitivity to Data Uncertainty . . . 259
8.1.1 Uncertainty in Data . . . 259
8.2 Towards a Massively Parallel Implementation . . . 266

**9 Conclusions** **275**

**A Algorithm Pseudo Code** **284**
A.1 Pseudo Code for `HeightAbove()`. . . . 284
A.2 Pseudo Code for `INFLOWS()`. . . . 286
A.3 Pseudo Code for `ALLFLOW3()`. . . . 287

# List of Figures


2.1 A graphical depiction of an undirected graph. . . . . 34
2.2 A graphical depiction of a hypergraph. . . . . 36
2.3 Graphical depiction of message-passing using construction sequence (a) hypertree (b) Markov tree representation. . . . . 41
2.4 Graphical depiction of message passing using a Markov tree representation. . . . . 43
2.5 Graphical depiction of a directed acyclic graph (DAG). . . . . 44
2.6 Mechanisms of Influence. . . . . 47
2.7 Effect of Quantisation Errors. . . . . 52
2.8 Ridge and Valley Descriptions. . . . . 61
2.9 Terrain and Associated Topography Tree. . . . . 61
2.10 Arming Method. . . . . 62

3.1 Relative location of the two study areas. . . . . 78
3.2 The Moora / Kalannie-Goodlands study area. . . . . 80
3.3 Landsat TM enhancement of the MKG study area, with bands 2, 3 and 4 in blue, green and red. . . . . 82
3.4 Average annual rainfall and associated weather station sites. 420mm (red) through to 270mm (green). . . . . 83
3.5 Digital elevation model derived from broad-scale elevation data. Black denotes relatively low areas and white denotes relatively high areas. . . . . 85

3.6 Relative locality of the Upper Kent River Catchment. . . . . 88

4.1 Standard Algorithm. . . . . 94
4.2 Standard data processing scheme. . . . . 96
4.3 Incorporating the binary search tree. . . . . 106

5.1 Sample sites for the accurate elevation data (area approx. 4km. by 4km.). . . . . 116
5.2 DEM derived from accurate elevation data. The colours range from yellow, depicting the areas with the lowest elevation, through green, blue, purple, pink and then red, which depicts the regions with the greatest elevation. . . . . 117
5.3 Sample sites for broad-scale elevation data. . . . . 118
5.4 DEM derived from broad-scale sites. The colours range from yellow, depicting the areas with the lowest elevation, through green, blue, purple, pink and then red, which depicts the regions with the greatest elevation. . . . . 119
5.5 Difference image of the DEM derived from the broad-scale data and the *true* elevation. . . . . 120
5.6 Three dimensional perspective of the top left hand quadrant of the image in figure 5.5 . . . . . 121
5.7 Contour intervals extracted from the *true* elevation. The broad-scale elevation data is sampled at elevations depicted by the black contour lines, which when compared with the two metre contour lines shown in red, demonstrates that elevations may be under estimated for hilltops and ridges and over estimated for lower lying areas such as valleys. . . . . 122
5.8 Landform Partitioning using MLC of Slope and Curvature Variables. 124

5.9 Drainage network, in blue, resulting from the GRASS routine r.watershed. The true drainage network is shown in red while the GRASS derived watershed boundaries are shown in yellow. . . . . 126
5.10 DEM partitioning using `HeightAbove()`. . . . . 129
5.11 The partitioning shown in figure 5.10 overlaid on the DEM to give a three dimensional perspective. Note that classes less than 1m above the nearest drainage pixel lie in the lower parts of the landscape whereas those greater than 10m above the nearest drainage pixel tend to be hill tops or ridges. . . . . 129
5.12 A graphical depiction of a limitation of the `HeightAbove` algorithm. This figure depicts a cross-sectional view of a ficticous landscape. In this figure, the segment of the thick black line which falls within the watershed of $V_1$ will have *height above* values calculated relative to $V_2$, whereas values relative to $V_1$ are required. . . . . 132
5.13 Three dimensional depiction of the results obtained from the single-flow accumulation algorithm. Here the accumulation image has been draped over the broad-scale DEM. The figure is coloured red, representing the areas of highest accumulation, through blue, cyan, green and yellow, which depicts the regions of lowest accumulation. 133
5.14 Three dimensional depiction of the results obtained from the multiple-flow accumulation algorithm. Here the accumulation image has been draped over the broad-scale DEM. The figure is coloured red, indication the areas of highest accumulation, through blue, cyan, green and yellow, which depicts the regions of lowest accumulation. 133
5.15 Flat area classification of the broad-scale DEM produced using the algorithm described by Chorowicz et al. (1989). Flat hill tops and ridges, flat valleys, flat inflextions and non-flat regions are coloured green, red, blue and white respectively. . . . . 137

5.16 Three dimensional depiction of the flat area classification of the broad-scale DEM produced using the algorithm described by (Chorowicz et al. 1989). The colours are the same as for figure 5.15. . . . . 138
5.17 New feature characterisations: (left) broad valley and (right) flat hilltop. The grey region represents the feature of interest and the positive and negative symbols indicate that a bounding pixel is higher than or lower than the feature respectively. . . . . 139
5.18 Results of the INFLOWS algorithm. Slope = 0.01. The regions are coloured green for $R < 0.1$, through to yellow, orange and then red for $R > 0.9$. . . . . 142
5.19 Results of the INFLOWS algorithm. Slope = 0.02. The regions are coloured green for $R < 0.1$, through to yellow, orange and then red for $R > 0.9$. . . . . 143
5.20 Results of the INFLOWS algorithm. Slope = 0.03. The regions are coloured green for $R < 0.1$, through to yellow, orange and then red for $R > 0.9$. . . . . 144
5.21 A three dimensional perspective of the results of the INFLOWS algorithm (slope = 0.02). The INFLOWS results are draped over the DEM. . . . . 144
5.22 Results of the combined partitioning. Here the colours yellow, green, cyan, blue and red represent the classes HillTops, Ridges / UpperSlopes, UpperValleys, Lower Valleys and BroadValleys respectively. . . . . 145
5.23 A flat region, coloured grey, with a potential flow feed-back problem. In the diagrams, bounding pixels which are higher that or lower than the nearest flat feature pixel are labelled with the symbols '+' and '-' respectively. The flow characteristics of this region (left) may result in flow feed-back path shown by the arrowed path (right). . . . . 148

5.24 Example of ALLFLOW3 Partitioning. . . . . 150
5.25 Example of ALLFLOW3 Partitioning: three dimensional perspective view. . . . . 151
5.26 Partitioning of the MKG Study Area. . . . . 151

6.1 Image of soil types. . . . . 161
6.2 Salinity ground-truth: SaltAffected (red), RiskOfSalinity (yellow) and NoRisk (green). . . . . 162
6.3 Classification produced by MLC. . . . . 164
6.4 Simple Models: (a) MLC; (b) MLC2; (c) MODEL1; and (d) MODEL2. 164
6.5 Classification produced by MLC2. The colour key is given in figure 6.3. . . . . 166
6.6 MODEL1–EM classification. The colour key is given in figure 6.3. . 171
6.7 MODEL2–EM classification. The colour key is given in figure 6.3. . 172
6.8 MODEL2–OBV classification output. The colour key is given in figure 6.3. . . . . 175
6.9 Temporal Models. . . . . 186
6.10 A graphical depiction of the neighbourhood model proposed by Besag (1986). . . . . 189
6.11 Temporal / Spatial Models. . . . . 191
6.12 Model Comparison (1990): (top-left) MLC, (top-right) MLC-NBR, (middle left) MODEL4, (middle right) MODEL6 and (bottom) MODEL6*. . . . . 195
6.13 Soil information grouped into three classes. . . . . 206
6.14 Probability of a soil change ($P(SoilChange)$), white = yes, black = no. . . . . 206
6.15 Probability of a convergent flow point ($P(ConvergentFlow)$), white = yes, black = no. . . . . 207
6.16 Stratified *Water Accumulation* image. . . . . 207
6.17 Flat Area Classification. . . . . 208

6.18 Salinity Ground Truth. . . . . 208
6.19 Graphical depiction of the network before training. Here variables are drawn as tables, and the variables' prior probabilities are typed in bold font. . . . . 212
6.20 Graphical depiction of the network after training. Note the change (compare figure 6.19) in the variables' prior probabilities. . . . . 216
6.21 Network output: Salinity map. *Saline* areas are coloured red and *Normal* areas are coloured green. Note that *PoorProductivity* is never the most likely label. . . . . 218
6.22 Network output: $P(Salinity)$. The map is coloured red, which indicates a high probability of salinity, through blue, cyan, green, yellow and then white which indicates a low probability of salinity. . 218

7.1 Result of `INFLOWS` algorithm. The figure is coloured red, which indicates a high INFLOWS value, through pink, purple, blue, cyan, green and then yellow, which indicates a low INFLOWS value. Pixels that are white are not flat. . . . . 231
7.2 Result of the `FLOWSLOPE` algorithm in greyscale. White and black indicate high and low slope respectively. . . . . 232
7.3 Result of the `ALLFLOW3` algorithm. The figure is coloured red, which indicates a high accumulation value, through pink, purple, blue, cyan, green and then yellow, which indicates a low accumulation value. . . . . 233
7.4 Buffered Convergent Flow Points. . . . . 234
7.5 Upslope Cleared area. The figure is coloured red, which indicates a relatively large upslope cleared value, through pink, purple, blue, cyan, green and yellow, which indicates a relatively low upslope cleared value. . . . . 236

7.6 Percentage upslope cleared. The figure is coloured red, which indicates 100% upslope cleared, through pink, purple, blue, cyan, green, yellow and white, which indicates a minimal percentage of upslope clearing. . . . . 237

7.7 Expected Ground–Water Salinity. Red, green and blue regions have a high, medium and low expected groundwater salinity respectively. 238

7.8 Location of training and validation data, coloured blue and red respectively . . . . . 239

7.9 Proposed model. . . . . 239

7.10 Network output: most likely class label, 1988. *Saline*, *PotSaline* and *NoRisk* areas are coloured red, blue and green respectively. . . 242

7.11 Network output: neighbour-modified results 1988. *Saline*, *PotSaline* and *NoRisk* areas are coloured red, blue and green respectively.243

7.12 Catchment summary, neighbour modified results. The colours green, cyan, blue, red and grey identify land which was mapped as *Saline* in 1977, 1988, 1994, *PotSaline* in 1994 and *NoRisk* until at least 2004 respectively. . . . . 251

8.1 Landsat TM classification. Of interest here is the *BareSalt* class, which is coloured green. The other classes are *Water*, *RemVeg*, *BareSoil*, *WaterSideRemVeg*, *AgrLand* coloured red, blue, cyan, yellow and pink respectively. . . . . 260

8.2 Attribute uncertainty in *BareSalt* class ($P(BareSalt)$). The figure is coloured red, which represents a high probability, through pink, blue, cyan, green and yellow, which represents a low probability. White represents $P(BareSalt) = 0$. . . . . 260

8.3 Probability of *BareSalt* derived from applying the Gaussian filter to the *BareSalt* class of figure 8.1. The figure is coloured red, which represents a high probability, through pink, blue, cyan, green and yellow, which represents a low probability. White represents $P(BareSalt) = 0$. . . . . 261

8.4 Probability of *BareSalt* including positional and attribute uncertainty. The figure is coloured red, which represents a high probability, through pink, blue, cyan, green and yellow, which represents a low probability. White represents $P(BareSalt) = 0$. . . . . 262

8.5 Difference of the images shown in figures 8.3 and 8.4. The figure is coloured red, representing a relatively large positive difference of approximately 0.25, through pink, blue, cyan, and green, which represents a relatively large negative difference of approximately 0.15. White represents no differences. . . . . 262

8.6 *Fuzzy* rule generation as defined in equation 8.1. . . . . 265

8.7 Salinity Ground Truth (1994). Saline (green), non-saline (red). . . . 267

8.8 Network output — $P(NoRisk)$, (1994), without data uncertainties included. The figure is coloured red, representing a high (low) value for $P(NoRisk)$ ($P(SalinityRisk)$), through pink, blue, cyan, green and yellow, representing a low (high) value for $P(NoRisk)$ (($P(SalinityRisk)$). . . . . 267

8.9 Network output — $P(NoRisk)$, (1994), with data uncertainties included. The figure is coloured red, representing a high (low) value for $P(NoRisk)$ ($P(SalinityRisk)$), through pink, blue, cyan, green and yellow, representing a low (high) value for $P(NoRisk)$ (($P(SalinityRisk)$). . . . . 268

8.10 Network output — $P(NoRisk)$, (1994), where black represents areas which are *sensitive* to data uncertainty. . . . . 268

8.11 Results Using Compacted Parameters. . . . . 274

# List of Tables


4.1 Algorithm performance - number of sub-problems generated and algorithm processing times. . . . . 103

4.2 Algorithm performance - solution sizes. . . . . 103

5.1 Relationship between HeightAbove and Salinity. . . . . 130

5.2 Relationship between HeightAbove and Salinity. . . . . 130

5.3 Comparison of the sensitivity of the single-flow and multiple-flow algorithm to errors in the DEM. . . . . 135

5.4 Comparison of the single-flow and multiple-flow algorithm to poor elevation definition in the broad-scale DEM. . . . . 136

5.5 Coincidence table of salinity information and the flat-valley segments generated using different slope thresholds. The numbers are pixel counts. . . . . 141

5.6 Cross tabulation of salinity and the landfrom classes defined using the approach presented in section 5.4.2. . . . . 146

5.7 The relationship between salinity and landform as derived from the counts given in table 5.6. . . . . 146

5.8 Cross tabulation of salinity and the landfrom classes defined using the ALLFLOW3 algorithm. . . . . 152

5.9 The relationship between salinity and landform as derived from the counts given in table 5.8. . . . . 152

6.1 Diagonal classification error estimate summary, $p(l_i^*|l_i)$, for each year $i$. . . . . 165
6.2 Parameter estimates derived by OBVIOUS means. . . . . 169
6.3 MODEL1-EM parameter convergence. . . . . 170
6.4 MODEL2-EM parameter convergence. . . . . 170
6.5 MODEL1–EM parameter estimates. . . . . 170
6.6 MODEL2–EM parameter estimates. . . . . 171
6.7 Model/classifier comparisons. . . . . 174
6.8 Paddock-use mapping accuracies, 1989. . . . . 177
6.9 Paddock-use mapping accuracies, 1990. . . . . 178
6.10 Paddock-use mapping accuracies, 1991. . . . . 179
6.11 Paddock-use mapping accuracies, 1992. . . . . 180
6.12 Average paddock-use accuracies (%), 1989–1992. . . . . 180
6.13 Summary of the spatial uniformity improvements obtained from each of the models. . . . . 182
6.14 Accuracy for salinity classification. . . . . 193
6.15 Landcover Accuracies, 1989. . . . . 199
6.16 Landcover Accuracies, 1990. . . . . 200
6.17 Landcover Accuracies, 1991. . . . . 201
6.18 Landcover Accuracies, 1992. . . . . 202
6.19 Average Paddock-Use Accuracies (%), 1989–1992. . . . . 202
6.20 Improvement in Spatial Uniformity (%), 1989–1992. . . . . 202
6.21 Farm–paddock transition characteristics. For each farm the first of the two rows give the transition counts (in number of paddocks) and the second of the two rows the associated transition probabilities. 210
6.22 Network Mapping Accuracies: Network vs Technician. . . . . 217
6.23 Network Mapping Accuracies: $P(Technician|Network)$. . . . . 217
6.24 Network Mapping Accuracies: $P(Network|Technician)$. . . . . 217

7.1 Mapping accuracy over training data. . . . . 244

7.2 1994 *Saline* training sites within $X$m and $H$m height of network prediction. . . . . 247
7.3 Neighbourhood modified results. . . . . 248
7.4 Mapping accuracies over validation data. . . . . 250
7.5 Neighbourhood modified mapping accuracies over validation data. . 250
7.6 Catchment summary. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*. . . . . 252
7.7 Catchment summary — neighbour-modified results. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*. . . . 252
7.8 Cleared land catchment summary. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*. . . . . 253
7.9 Remnant vegetation catchment summary. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*. . . . . 253
7.10 Cleared land catchment summary — neighbourhood-modified results. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*. . . . . 253
7.11 Remnant vegetation catchment summary — neighbourhood-modified. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*. . . . . 253

8.1 Results of Network Parameter Compaction. . . . . 272

# Chapter 1

# Introduction

The motivation behind the research reported in this thesis is to develop methodologies and working models for the improved monitoring of land condition in the south western agricultural region of Western Australia, east of the Darling scarp. This area represents some 1.6 million hectares of land, predominantly used for grazing in the higher rainfall areas (> 500mm. average annual rainfall) in the south–west of the region, and for the growing of cereal grains and grazing in the drier (300mm. average annual rainfall), north-eastern part of the region.

From a monitoring perspective, issues of interest include the loss of previously productive land to salinity, the loss of valuable top soil by wind and water erosion, the loss of productivity of pastures and cereal crops due to waterlogging, general crop and pasture productivity and the (potentially changing) condition of remaining remnant vegetation covers.

Because of the size and extent of the wheatbelt, and its typically low level of productivity, cost-effective broad-scale monitoring techniques are desirable and hence remote sensing (RS) and geographical information system (GIS) technologies are typically employed.[1]

Remote sensing systems provide digital, broad-scale data relating to various

[1] Computer-based techniques have the advantage over manual techniques in that they are cost effective requiring less human labour than traditional (human) interpretation techniques.

features of the land. For example, satellite sensors including Landsat TM and Spot provide optical data collected at various wavelengths which give information on landcovers; sensors based on microwave technology may provide information on land surface topography; and sensors based on electromagnetic conductance may provide information on subsurface phenomema.

A GIS is an integrated computer system for handling and analysing spatial data; generally providing tools for collecting, storing, retrieving, transforming, analysing and displaying spatial data. GIS are used for a wide range of tasks including planning and monitoring of large spatial problems/events.

Mapping applications within this domain include using remotely sensed Landsat TM data to estimate the extent of salinity (Wallace and Wheaton, 1990; Wheaton et al., 1992), waterlogging of cereal crops (Wallace et al., 1993), disease (dieback) in remnant vegetation (Behn and Campbell, 1992) and wind erosion (Palmer et al., 1994). Although the use of remote sensing is not new to mapping and monitoring land condition within the state, it has some limitations which mainly result from errors associated with using single-date imagery. A combination of multi–temporal satellite data and GIS data would be expected to reduce the number of errors made.

Many different approaches to combining data exist. A common approach is to apply *if–then–else* rules to a number of data layers to form a resulting interpretation. In such approaches, errors in data and the resulting interpretations are often ignored. The wheatbelt provides an interesting area when considering the combination of data as many data are collected for reasons other than for the specific task at hand, or they have been collected on a value-for-money basis and typically do not have the resolution/quality which could be hoped for. In this regard, much of the data contain an element of error and their use has a degree of uncertainty associated with them. Further, because some (potentially important) data layers are typically not available, then any decision made on the basis of the available data will contain an element of uncertainty.

With the above points in mind, this thesis considers the problem of using computer-based approaches to combine multiple sources of error-prone data with the aim of improving estimates in the condition of land cover, and in particular the mapping, monitoring and prediction of salinisation.

Existing techniques which are commonly employed for the interpretation of remotely sensed and other data are reviewed in section 2.2. The potentially large rule sets required for combining many sources of data have lead many researchers to use *expert system / knowledge–based system* technology for rule base management. Such systems offer approaches for rule set creation, interrogation, modification and maintenance. A feature of the techniques described in section 2.2 is that they do not consider combining data in the presence of uncertainty, which is specifically considered here.

Various schemes for combining data in the presence of uncertainty have been proposed, and have been at the centre of much debate in the artificial intelligence community as well as the remote sensing and GIS communities. In applications involving remote sensing and GIS, much confusion between the virtues of the different methods was encountered in the literature. In section 2.3, the different approaches are described in detail, and, based on the properties of the different approaches, a method of combining data using probabilistic methods is adopted. The approach is based upon relatively new developments in the field of *Bayesian networks*, which allow for the creation of complex reasoning models and which are the basis of current *probabilistic expert/knowledge–based* systems.

Given a general scheme for combining data, an important aspect in regards to the accuracy of the resulting interpretation is that the appropriate data are combined. Satellite data such as Landsat TM are considered to be an important data set, providing information on landcover at regular intervals in time[2].

[2]Much work on the classification (and spectral characteristics) of remotely sensed data using techniques described by Campbell and Wallace (1989) has been previously performed, for example (Wallace and Wheaton, 1990; Wheaton et al., 1992; Wallace et al., 1993; Behn and Campbell, 1992) and considered to be well understood, and although remote sensing inputs are used in this research, their specifics are not discussed in detail in this thesis.

Digital elevation models promise to be another important data source, although they have received little attention in previous applications in broad–scale land condition monitoring in W.A.

While using elevation values directly may prove useful in a small region, the extraction of more generic landform features such as *hilltops* and *valleys* appears useful for broad-scale monitoring, as they are invariant to a mean shift in elevation from one region to another. The landscape in question provides a challenging area for the extraction of features, as the quality of the available elevation data is rather poor given the very flat characteristics of the terrain. For these reasons, published computer-based techniques for extracting landform features from DEMs are reviewed in section 2.5.

Two very different regions of the agricultural area were considered and are described in Chapter 3. The first area, described in section 3.2, is situated in a lower rainfall region of the agricultural area and is predominantly used for the growing of cereal grains and the grazing of sheep. This region is part of the Land and Water Resources Research and Development Corporation (LWRRDC) funded project, *Detecting and Monitoring Changes in Land Condition Through Time Using Remotely Sensed Data*, and hence the research emphasis on this area is focused on the mapping of salinity through time[3]. The second area, described in section 3.3, is situated in a relatively high rainfall area of the agricultural region and is predominantly used for growing pasture for grazing. This region is part of the LWRRDC funded project[4], *Integrating Remotely Sensed Data With Other Spatial Data to Predict Areas at Risk from Salinity*, and hence the emphasis on work in this area focused on salinity prediction.

Because of the limited availability of software for combining data using probabilities it was necessary to develop suitable programs. The combination algorithm and its implementation used for experiments presented in this thesis is described

[3]The results of this project are reported in Furby et al. (1995)
[4]The results of this project are reported in Evans et al. (1995)

in Chapter 4. The general combination scheme is based on the work of Lauritzen and Spiegelhalter (1988), which is covered in the review presented in section 2.3. Computational time is expected to be of major importance in applying the combination scheme to the vast quantities of data typical in remote sensing and GIS applications, and is addressed in two ways. Firstly, in section 4.3, an algorithm which addresses the most critical step in forming a computationally efficient network having general structure (ie. obtaining a good network *triangulation*) is examined and its performance is compared with an existing heuristic approach.

Secondly, in Section 4.4, a new scheme for improving prototyping times is introduced. The scheme restricts data to categorical form and relies on the fact that various data vectors are encountered many times. Results are obtained more quickly than from the general scheme, although some errors associated with the data are ignored. Rule uncertainty is still considered.

Another issue that arises with the use of probabilistic networks is the way in which the (potentially many) rule weights are obtained[5]. One approach is to have human experts give their intuitive feeling for what the rule weights may be and then use these numbers directly. Where multiple experts are consulted, then some means of pooling the results needs to be devised. This approach seems most appropriate where there are little data for estimating the parameters directly.

A problem with the above approach is that it can easily lead to a network with probabilities that are intuitively in conflict with what would be expected (Osherson et al., 1994). This result reflects the fact that knowledge acquisition must be performed with care in such circumstances, and is a statement that modellers sometimes have difficulty in deriving good probability estimates from experts.

Where much data are available, using the data to estimate the parameters is appropriate. An algorithm known as the Expectation Maximisation (EM) algorithm is considered in section 4.5.

Chapter 5 describes experiments in extracting features from broad-scale ele-

[5]This issue also arises with other schemes which represent uncertainty by weighted rules.

vation models. To understand errors likely to be encountered in the broad-scale data, and the effect that these may have on feature extraction, the DEM derived from broad–scale elevation data was compared with a DEM derived from *accurate* elevation data defined for a small region, and the discrepencies were recorded. Existing techniques were then applied to the broad-scale data and, due to the flat nature of the terrain, performed poorly. Based on the relative strengths and weaknesses of existing approaches, new approaches were devised and tested, and shown to provide a useful broad-scale partitioning of the terrain.

Given the means for extracting relevant information from remotely sensed data (Maximum Likelihood Classification or MLC for short) and DEMS (refer to chapter 5), and a procedure for combining multiple sources of data (chapters 2.3, 4), chapter 6 then describes some experiments in combining data for the purpose of improving land-cover/land-use assessments.

Firstly, in section 6.3, remotely sensed data for a single date are combined with a single source of ground data. These experiments build upon a relatively simple model for combining data first proposed by Strahler (1980). Two new extensions are made to Strahler's original model: (1) MLC classification error estimates are incorporated into the model; and (2) the EM algorithm is used to estimate the model parameters (rule weights). The success of the resulting interpretations are judged on the basis of accuracy in salinity mapping, paddock use mapping and on their spatial coherency (or "smoothness").

While some advantages of the new approaches are noted, some limitations are also exposed. The main weakness of the simple models is their failure to produce temporally consistent results for land cover classes which are known not to vary greatly for the time period considered. For this reason, simple temporal models are considered in section 6.4. The models combine a temporal sequence of Landsat TM data obtained for consecutive years. Like the models in section 6.3, the EM algorithm is used to estimate model parameters and a model incorporating MLC error rates is proposed. The success of the models was judged by the same criteria

as those presented in section 6.3, and shown to improve results by all three criteria. The benefits of incorporating prior knowledge (such as that gained from experts) in the application of the EM algorithm is also examined.

While the above models improve the spatial coherency of resulting interpretations, greater improvements would be expected by explicitly incorporating the spatial relationships of neighbouring pixels, ie. to include some mechanism for encouraging a pixel's label to be the same as the label of surrounding pixels. A new approach for incorporating spatial operations for general Bayesian networks is considered in section 6.4.2 and examples of its appliciation given. In the examples spatial operations were incorporated into the temporal models previously considered. Further improvements in mapping accuracy are obtained.

The above approaches have the following merits:

1. they are conceptually relatively simple;

2. they are mostly automated in their application[6]; and

3. they improve land-cover classification.

In section 6.5, the problem of mapping salinity is explicitly considered. Drawing from knowledge gained in sections 6.3 and 6.4, it was observed that ground data as well as temporal Landsat TM data are required for accurate salinity classification. Temporal Landsat TM data, data derived from DEMs (chapter 5), and soils information are combined. A knowledge-based approach is used to define the structure of the model and define initial rule weights. The EM algorithm is then used to refine the weights. The accuracy in salinity mapping is used to gauge the performance of the network. This section also demonstrates the form of information produced when used in an interactive mode.

In Chapter 7, the problem of broad-scale salinity prediction is investigated for the Upper Kent River Catchment. The catchment differs significantly from

[6]Estimation of the spectral signatures of the different landcover classes takes some human effort. However, once the spectral signatures have been defined, the application of the models is largely automated.

the Moora/Kalannie—Goodlands region, having much higher rainfall, different geology, different land use and a lesser amount of clearing of remnant vegetation.

This study was part of the LWRRDC project *Predicting Areas At Risk Of Salinity Using Remotely Sensed and GIS Data.* The overall project was a collaborative effort between a number of agencies, including the CSIRO, the Department of Land Administration (DOLA) and Agriculture Western Australia.

The aim of the project was to see if computer-based techniques could replicate a human expert in the mapping and prediction of salt-affected areas. Firstly, a forum of people including experts in hydrology and land care was held to establish which factors were likely to influence the salinisation of land. Based on these factors, available broad-scale data to be used for the predictions were collected and a GIS database was formed. Ground data were collected by an expert, to be used for parameter estimation and also validation.

The techniques developed in section 5.5 were used to form landform variables from the DEM, and MLC was used to form variables relating to landuse. Unfortunately, not all factors cited by the experts (such as subsurface phenomenon) as being important could be represented, as some data were not available.

A network for representing the relationships (and combining the data) is then conceived. In the first instance, some effort is made to extract rules and rule weights from the experts. This approach is initially taken, as only limited ground data is available; by defining rules (and associated weights), the lack of data can be partially offset by what can be inferred from the rules. However, some difficulties are experienced using this approach and hence a greater amount of ground data were collected, allowing the rule weights to be estimated directly. (Because of the different characteristics of the Kent region, the different data available and the different desired outcome, the structure of the network varied somewhat from that conceived for salinity mapping in section 6.5.)

The data are then combined using the network to produce a classification of the historical spread of salinity in the catchment and a prediction of its likely extent

in the future. The resulting interpretation is compared to independent ground truth.

# Chapter 2

# Literature Survey

## 2.1 Introduction

Remote sensing and geographical information system (GIS) technologies have been shown to form the basis for cost-effective, broad-scale assessments of many environmental monitoring problems including landuse and land condition assessments (Wilkinson et al., 1992; Kontoes et al., 1993; Wharton, 1987; Janssen and Middelkoop, 1992), forestry management (Skidmore, 1989; Behn and Campbell, 1992) and mineral exploration (Kruse et al., 1993; Moon, 1990).

From a remote sensing perspective, the interest generally is in quantifying the information that a specific sensor can provide and then using the data to provide information over a broad area. Techniques, such as described by Campbell and Wallace (1989); Kiiveri and Campbell (1992), that are specific to one form of data, are typically used, and results are expressed as maps of labels. In the RS literature, this form of interpretation is called *classification* and is reviewed in section 2.2.

One specific data set may not be sufficient for a given application; multiple data sets may have to be considered. GISs provide tools for managing and analysing spatial data, which include remotely sensed data. When used as a monitoring tool, a problem that naturally arises is "under what scheme should data be combined?". This problem is generally referred to as the *multi-source data combination problem*

and is introduced in section 2.2.3.

GIS typically provide tools to perform basic algebraic manipulations of data such as map intersections and unions. Logic-based expert systems provide an extension to this idea by providing a framework for expressing and manipulating logical relations (typically called *rules*). In this way, potentially complex sets of rules can be constructed, maintained, interrogated and applied to data. When such a set of rules is defined within the context of an expert system, it is typically called a *knowledge-based system.*

Typically the quality of the data available is governed by cost constraints and as such may not be ideal in terms of identifying features of interest. In such cases, the information derived from the data will have a degree of uncertainty associated with it, which ideally should be considered when combining data. Knowledge-based systems which incorporate uncertainty therefore appear advantageous.

Data uncertainty may take two forms: positional uncertainty and attribute uncertainty. Positional uncertainty occurs when data are displaced from their real co-ordinates, whereas attribute uncertainty arises from errors in the measuring system that is used to form class labels. Typically, GIS and remote sensing data will have an element of both forms of uncertainty.

How to incorporate uncertainty within expert systems has been at the centre of much debate in the AI, remote sensing and GIS literature (Lee, 1988; Henkind and Harrison, 1989; Lee et al., 1987) when the objective is to combine data. Two techniques commonly applied for combining data in the presence of uncertainty are those based on probability theory and the Dempster-Shafer Theory of Evidence (Shafer, 1976). Debate typically discusses the virtues of the calculus of each, but sometimes this debate extends into other issues which are artifacts of a particular implementation, and not shortcomings of the calculus in general. Both theories are reviewed in section 2.3, from which it is observed that both theories may have similar implementations and hence much of the debate may be diffused.

Probability theory, in particular Bayesian networks (Lauritzen and Spiegelhal-

ter, 1988), was chosen as the modelling environment. Such models are typically represented in a graphical form, giving the user a conceptual feel for what the model does. Graphical notation is introduced in section 2.3.4 and links to pre–existing data combination schemes are examined.

In the second half of the chapter, in section 2.5, methods for extracting features from digital terrain models are reviewed. Terrain has a significant influence in the management of land and larger-scale processes such as salinisation, although it has not previously been considered within Western Australia for broad-scale mapping and monitoring problems.

## 2.2 Classification

### 2.2.1 Background

The aim of classification is to recognise the state of a physical process by using a set of measurements recorded on it. The method used for performing the classification is generally called a *classifier*, while the recorded measurements are refered to as *data*. The state of the process recognised by the classifier is labeled as belonging to a particular *class*. After this stage, the labelled data are called the *classification* and the data are said to have been *classified*.

A typical classification problem is as follows: knowledge exists about $k$ possible classes $C = \{c_1, \ldots, c_k\}$ and $m$ features are recorded on the system, which can be represented as a vector $\underline{x} = \{x_1, \ldots, x_m\}$. With this information, the classifier decides to which class $c_i \in C$ a new data vector $\underline{x}$ belongs.

The process of classification requires at least the following steps:

1. defining the $C$ possible classes;
2. choosing a model for assessing the information in the available data; and
3. deciding on the class label after having assessed the information in the data.

When the available data uniquely specify the class to which an observation should belong, the assignment of the appropriate class label is made unambiguously; that is with absolute certainty. When the data do not uniquely identify a class, then some element of doubt about the chosen label will exist, or in other words the label will be chosen with an element of *uncertainty*.

Some methods of classification explicitly quantify the degree of uncertainty and others do not (see section 2.2.3). Probability theory provides one means of quantifying uncertainty, but others also exist. This point will be discussed further in section 2.3. In this respect, a method of classification is governed by an underlying model, which will be the terminology used in the remainder of this thesis.

Different models may work better with some forms of data than others. For example, multivariate Gaussian distributions are typically used for modelling land-cover classes for the classification of optical remotely sensed data, but would not be suitable for classifying combinations of categorical data say (for which estimates of the covariance structure of the model would not be defined). If the classification task requires many different forms of data to be used, then combining different existing models to interpret the data may be appropriate. In this respect the new model formed is often referred to as a *joint* model.

Classifiers can be grouped into two types: *supervised* and *unsupervised* classifiers, depending on the degree to which the classifier is trained.

### 2.2.2 Processing of Remotely Sensed Data

This section considers the classification of remotely sensed spectral data considered in isolation (that is without taking into account other data sets). These data are available from satellites such as Landsat or Spot (or from airborne scanners), and typically consist of a number of reflectance measurements at different wavelengths for each image pixel.

The most popular classifier for this form of data is the Maximum Likelihood

Classifier (MLC) (Rao, 1966). Given $m$ spectral bands, these classifiers generally assume that spectral descriptions for the $k$ classes $l_i$, $i = 1, \ldots, k$ can be modelled using multivariate Gaussian densities $f_i(\underline{x}_m, \bar{\underline{x}}_i, \underline{V}_i)$ defined as

$$f_k(\underline{x}_m, \bar{\underline{x}}_k, \underline{V}_k) = (2\pi)^{-m/2}|\underline{V}_i|^{-1/2}e^{-1/2D^2_{i,m}} \quad (2.1)$$

$$D^2_{i,m} = (\underline{x}_m - \bar{\underline{x}}_i)^T \underline{V}_i^{-1}(\underline{x}_m - \bar{\underline{x}}_i) \quad (2.2)$$

where $\bar{\underline{x}}_i$ is the vector of sample means, $\underline{V}_i$ is the sample covariance matrix, $\underline{x}_m$ is the pixel data vector and $D^2_{i,m}$ is the squared Mahalanobis distance for vector $\underline{x}_m$ and class $l_i$.

The quantities $\bar{\underline{x}}_i$ and $\underline{V}_i$ may be estimated from the data by specifying samples of data for each class. This specification is generally performed by an expert. In the following, sample data that are used to estimate model parameters will be called *training* data.

The classifier uses the densities to calculate the probability that a pixel belongs to a particular class, and then typically assigns that pixel to the class having the maximum probability of class membership.

More specifically, assuming multivariate Gaussian densities for $k$ classes and an image composed of $m$ spectral bands, a pixel is assigned to the class with label $l_*$ such that:

$$P(l_*|\underline{x}) = \max_{i=1,\ldots,k} P(l_i|\underline{x}_m) \quad (2.3)$$

$$\text{where } P(l_i|\underline{x}_m) = \frac{f_i(\underline{x}_m, \bar{\underline{x}}_i, \underline{V}_i)P(l_i)}{\sum_{z=1}^{k} f_z(\underline{x}_m, \bar{\underline{x}}_z, \underline{V}_i)P(l_z)} \quad (2.4)$$

For more information on the MLC applied to remotely sensed data, refer to the work by Campbell and Wallace (1989); Kiiveri and Campbell (1992).

Other popular techniques for the classification of remotely sensed data include

neural networks and decision trees. As this thesis focusses on building upon existing (operational) techniques, these approaches will not be considered further here.

### 2.2.3 Classification using Remotely Sensed and Other Data

To improve classification accuracies, data other than RS data may be required. By incorporating ancillary data such as soil maps and attributes relating to landform morphology, known relationships may be incorporated into the mapping process to remove some obvious errors encountered when using RS data in isolation. Many authors have taken such an approach in other applications, for example Janssen and Middelkoop (1992); Hutchinson (1982).

Classifying multiple sources of disparate data requires that the data be considered together under *some* framework, of which many have been suggested. The selection of one method in preference to another depends upon the nature of the problem and the data. For example, when error-free data which uniquely describe each class are available, methods which incorporate uncertainty are not required. If, however, some data are missing or contain errors, other approaches are necessary.

Different models found in the literature are described below.

A straight-forward approach to analysing multisource data is to form an extended data vector for a pixel, such that the data vector contains a component from each source, and then analyse the data in the same way as for a single source. This technique works well for analyses such as multitemporal/multispectral image classification, where each source can be modelled using a statistical model such as the MLC (equation 2.4). This technique may not be appropriate when data layers do not conform to the same statistical model, for example Landsat TM data and soil maps interpreted from aerial photographs.

For mixed data sources, other approaches have been devised. One approach is to stratify the data based on a subset of the sources and then to analyse each

stratum using the remaining sources. This approach has been used Strahler et al. (1979) to improve forest-cover classification by incorporating topographic data into the analysis of Landsat Multispectral data. The Landsat data were stratified according to elevation, separate density functions were specified for each stratum and then each stratum was classified (with a MLC) separately.

Another alternative is to perform a classification based on one or more of the data layers and use the remaining layers to resolve conflicts/ambiguities. This technique was used by Hutchinson (1982) to resolve, with the use of slope data, the spectral confusion found between bright surfaces of a dry lake bed and steep sunny slopes of sand dunes.

Application of this technique consists of a set of classified data which is reclassified by an *if–then–else* rule set, and will be referred to as *post classification label modification*, or PCLM for short. While the entire process is one of classification, the term *post classification* emphasises that the process is disjoint in the sense that class labels are firstly produced by the MLC and then these labels are modified according to some rules. With such a procedure, the pixel label probabilities produced by the MLC are often ignored, in which case the overall classification system makes no distinction between data (MLC labels) which are poorly defined as opposed to well defined.

Rules expressed in an *if–then–else* structure are the basis of many popular expert system shells, such as CLIPS and POPLOG. Such an expert system shell may provide an environment in which to develop and apply the rules. This approach has been taken, for example, by Ton et al. (1991); Wang and Newkirk (1988); Wang (1990); Kruse et al. (1993); Leung and Leung (1993).

Application of methods such as PCLM and their extensions, based on *if–then–else* rules applied to class labels, lose, at an early stage in the classification process, the concept that the labels produced from a technique such as MLC have uncertainty associated with them. In effect, a decision is made too early in the process. The remainder of the classification process treats all labels equally, no matter

whether they are well defined (low uncertainty) or very poorly defined (high uncertainty). In this regard these approaches may be more sensitive to data with errors and may not generalise well over larger areas where the quality of data (amount of errors) may vary.

It would appear advantageous to delay any decisions in the classification process for as long as possible, or in other words, to propagate the uncertainties of the process for as long as possible, making a decision at the end of the process. This notion has lead to researchers combining data using techniques that propagate uncertainty. Many different formalisms exist for representing and propagating uncertainty, and not surprisingly, many different applications based on the different formalisms have emerged. Some of the different approches are discussed in the next section.

## 2.3 Reasoning With Uncertainty In Knowledge-Based Systems

The two most common modelling methodologies used within this area are those based on probability theory and the Dempster–Shafer (Dempster, 1968; Shafer, 1976) (DS for short) theory of evidence. Probability theory has a long history as a tool for the representation of uncertainty. Here the Bayesian interpretation (Papoulis, 1985) of probability is taken, and is described in Section 2.3.1. The DS theory of evidence aims to quantify uncertainty by *degrees of belief.*[1] In the DS theory of evidence, belief is represented as a *mass* function and evidence is combined using *Dempster's rule of combination.* The theory was first introduced as providing a generalisation of Bayesian inference (Dempster, 1968), and hence has many formal similarities with probability theory. The main difference between the two approaches is that, unlike the Bayesian approach, the DS approach allows

[1] A description also now used for probabilities in probability-based expert systems.

evidence to be assigned to sets of hypothesis rather than only a single hypothesis in isolation. The DS theory of evidence is introduced in section 2.3.2.

In the general case, these methods are computationally intractible. There are two basic ways to reduce the computational burden:

1. Approximate the theory with a calculus that is computationally efficient and accept any contradictions with the base theory on which the scheme was devised. These models are generally called *heuristics.*

2. Impose or exploit structure of the domain when performing the computations.

Perhaps the most well known heuristic is based on *certainty factors* (Shortliffe and Buchanan, 1975) and was first used in the medical expert system MYCIN (Shortliffe, 1976). Certainty factors indicate to the user the two quantities *measure of increased belief* and *measure of increased disbelief.* The quantities are *relative* measures of change in belief, unlike say probabilities which are absolute and can have a tangible interpretation if a *relative frequency* interpretation is considered (see section 2.3.1). For example, systems built on relative change in belief may inform the user that the belief in an outcome has changed from 1x to 100x, indicating a hundred fold increase in belief (an outcome considered probable under these reasoning schemes). Schemes bases on absolute quantities can, however, inform the user that the outcome with a hundred-fold increase in belief is still relatively unlikely, for example the probability of an event $A$ occurring may have changed from 0.000001 to 0.0001. It is this last point which makes a scheme which is truly representative of probability theory attractive, especially for representing and summarising RS and GIS data, where summaries are quite often presented as relative frequencies (or percentages of areas). For this reason, heuristics will not be considered any further.

Many early implementations of the different reasoning calculi, whether heuristic or otherwise, rigourously imposed modelling constraints to enforce computa-

tional efficiency. A common modelling constraint was that variables could only have two outcomes (Shortliffe and Buchanan, 1975; Adams, 1976; Gordon and Shortliffe, 1985) and/or be arranged in simple *tree-like* structures allowing only simple *two-way* relationships to be explicitly modelled (Shortliffe and Buchanan, 1975; Gordon and Shortliffe, 1985; Pearl, 1986b).

More recent efforts (Lauritzen and Spiegelhalter, 1988; Shenoy and Shafer, 1990) have produced frameworks that allow more general relationships to be modelled, with the underlying philosophy being that in many cases computational efficiency may be achieved by exploiting the structure of a given model/rule-base, instead of imposing structure. Exploiting structure appears advantageous, as it suggests that computational efficiency is achieved without any (artificial) loss to the quality of the model.

Computational algorithms that exploit model structure exist for both probabilistic (Lauritzen and Spiegelhalter, 1988; Shenoy and Shafer, 1986, 1990) and DS (Shenoy and Shafer, 1986, 1990) methods. These schemes rely on the model being able to be decomposed into small subsets of variables. Computational efficiency may be obtained by performing calculations local to these small subsets rather than considering all variables at once. These algorithms are discussed in Section 2.3.3.

Before considering these models, however, some background on probability theory and the DS theory of evidence will be given in Sections 2.3.1 and 2.3.2 respectively, and some required graph theory is given in Section 2.3.3.

### 2.3.1 Background on Bayesian Probability Theory

Probability theory has had a long history as a tool for handling uncertainty and modelling partial beliefs. Interpreting probabilities as *relative frequencies* of events provides a natural, common-sense way of examining and scrutinising results.

The use of probabilities in the remainder of this thesis makes extensive use of the probabilistic concepts of *joint distribution, Bayes rule, conditioning* and

*marginalisation*, to be described below. In Section 2.3.3, a computational scheme for performing the calculations efficiently in complex domains is reviewed and in Section 2.3.4 a conceptual framework for these domains is described.

Generally we are interested in making a decision from a set of mutually exclusive, competing hypotheses.

Let $\Omega = \{\omega_i | i = 1 \ldots n\}$ represent the set of $n$ alternate hypotheses $\omega_i$, and $2^\Omega$ represent the power set of $\Omega$, that is, $2^\Omega = \{A | A \subseteq \Omega\}$ is the set of all subsets of $\Omega$.

Let $(\Omega, \Xi)$ be a sample space, where $\Omega$ is a finite set and $\Xi = 2^\Omega$. A real-valued set function $P : \Xi \rightarrow [0, 1]$ defined on $(\Omega, \Xi)$ is called a probability (function), if it satisfies the following conditions:

1. $P(A) \geq 0 \ \forall A \subseteq \Omega$
2. $P(\Omega) = 1$
3. For $A, B \subseteq \Omega$, from $A \cap B \neq \emptyset$ follows $P(A \cup B) = P(A) + P(B)$.

The relative frequency of an event $A$ occurring is given by:

$$P(A) = \frac{|A|}{|\Omega|} \geq 0 \ \forall A \subseteq \Omega. \tag{2.5}$$

Suppose that the variables $A$ and $B$ are discrete, taking values $a_i$ and $b_k$ with respective probabilities $P(A = a_i) = p_i$ and $P(B = b_k) = p_k$. Their *joint* statistics are determined in terms of the *joint probabilities* $P(A = a_i, B = b_k) = p_{i,k}$. The *marginal* probabilities, $P(A)$ and $P(B)$, are obtained by $P(A) = \sum_B P(A, B)$ and $P(B) = \sum_A P(A, B)$. This process is often referred to as *marginalisation.*

The Bayesian interpretation of probabilities assumes that numerical values quantifying the probability distribution for $A$ are collected relative to some underlying background knowledge, $K$, say. These distributions are called *conditional* probabilities and are denoted $P(A|K)$ and defined as

$$P(A|K) = \frac{P(A \cap K)}{P(K)}. \tag{2.6}$$

From the definition of the conditional probability of $A$, the probability for the intersection of the events $A$ and $K$ is obtained as follows

$$P(A \cap K) = P(K)\ P(A|K) \tag{2.7}$$

which can be generalised to obtain what is called the *chaining rule*. That is, given a sequence of $n$ events $A_1, \ldots, A_n \subseteq \Omega$ such that $P(\cap_{j=1}^{n-1} A_j) > 0$, then

$$P(\cap_{j=1}^{n-1} A_j) = P(A_1)\ P(A_2|A_1)\ P(A_3|A_2 \cap A_1)\ \ldots P(A_n|\cap_{j=1}^{n-1} A_j). \tag{2.8}$$

Two events $A$ and $B$ are said to be (statistically) *independent* if and only if

$$P(A \cap B) = P(A)P(B). \tag{2.9}$$

Two events $A$ and $B$ are *conditionally independent* given $C$ if

$$P(A \cap B|C) = P(A|C)P(B|C). \tag{2.10}$$

Another rule, called the *total probability rule*, states the following: let $H_1, \ldots, H_n$ be a partition of $\Omega$, that is $H_i \cap H_j = \emptyset$, if $i \neq j$ and $H_1 \cup \ldots \cup H_n = \Omega$. If $P(H_i) > 0$ for $i = 1, \ldots, n$, then

$$P(A) = \sum_{i=1}^{n} P(H_i)\ P(A|H_i) \tag{2.11}$$

holds for each event $A$. This rule leads to the well known *Bayes Theorem*, which is given below.

Let $H_1, \ldots, H_n$ be a partition of $\Omega$ such that $P(H_j) > 0\ \forall i = 1, \ldots, n$, and let

$A$ be an event with $P(A) > 0$. Then

$$P(H_j|A) = \frac{P(H_j)\ P(A|H_j)}{\sum_{i=1}^{n} P(H_i)\ P(A|H_i)}. \tag{2.12}$$

When probabilities are used within the Bayes interpretation, the members of the partition $H$ are sometimes called *hypotheses.*

Finally, consider the joint distribution of, say, three variables $A, B$ and $C$, written as $P(A, B, C)$. Observe that $C$ is in state $c_j$, that is $C = c_j$. Given this information, the joint distribution $P(A, B, C = c_j)$ may be more relevant. This process is refered to as *conditioning.* The process of conditioning and marginalisation is at the core of probabilistic expert systems. For example, if the expert system rule set is expressed as the (joint) distribution $P(A, B, C)$, then the initial belief of the system in the quantities $A$, $B$ and $C$ is given by $P(A)$, $P(B)$ and $P(C)$, which are obtained by marginalisation. If evidence that $C = c_j$, say, becomes available, then the rule set is *conditioned* to obtain $P(A, B, C = c_j)$ from which are obtained the revised beliefs $P(A|C = c_j)$ and $P(B|C = c_j)$ for $A$ and $B$, by marginalising the conditioned distribution.

In the following, a (possibly unnormalised) joint distribution specified for a set of variables $N$, say, will be denoted $\psi^N$. These distributions are often referred to as *potentials* (Lauritzen and Spiegelhalter, 1988).

In such systems, it is often convenient to express rules as conditional probability tables, from which we may obtain the required joint distribution by the chaining rule (equation 2.8). Bayes theorem (equation 2.12) is then used to manipulate the conditional distribution, from which the revised beliefs are obtained by marginalisation.

As will be seen in Section 2.3.3, these operations may be performed efficiently by exploiting the *structure* of the distribution, allowing calculations to be performed locally on smaller sets of variables. This scheme relies on *projection* (see Section 2.3.2) and *combination* of probabilities, the latter of which is defined below.

Given two potentials $\psi_1$ and $\psi_2$, the *combination* $(\psi_1 \oplus \psi_2)$ of the two potentials is given by:

$$(\psi_1 \oplus \psi_2)(A = a_i) = \psi_1(A = a_i)\ \psi_2(A = a_i)\ \forall\ a_i \in A. \tag{2.13}$$

### Probabilistic Models in GIS/RS

Combining other data sets with remotely sensed data may be achieved with simple extensions to the MLC; for example by considering a second data source (such as soil types) as providing information on the prior probability of observing a particular land cover (Strahler, 1980). That is, given a map of attributes, $D$ say, the MLC (equation 2.4) may be modified by replacing $P(C)$ with $P(C|D)P(D)$.

The above idea may also be used to add contextual information into the classification process (Kiiveri and Campbell, 1992). In this scheme, the distribution $P(C|D)$ represents the probability of obtaining a landcover at image location $(i, j)$ given the landcovers that have been observed at neighbouring locations $(i-1, j), (i+1, j), (i-1, j-1), (i+1, j-1), (i+1, j+1), (i, j+1), (i, j-1)$ and $(i-1, j+1)$. The validity of this approach is considered in section 6.4.2.

In Janssen and Middelkoop (1992), the same idea was applied to use temporal knowledge of crop rotations to improve classification accuracy. In their scheme, the distribution $P(C|D)$ represents the probability of observing a landcover type given observations of landcover types of previous years. Improvements in the interpretation accuracy were reported.

A Bayesian approach to the *multisource data combination problem* was considered by Lee et al. (1987). Their model assumes a set $X$ of $n$ sensors $X = \{x_1, x_2, \ldots, x_n\}$ which is used to provide information on a set hypothesis $W$. The data obtained from the sensors, $X$, is represented by the set of random variables $Y = \{y_1, y_2, \ldots, y_n\}$ where variable $y_i \in Y$ is associated with the data from sensor $x_i \in X$. Each sensor $x_i$ has an associated set of data classes $d_j \in D$ which have the interpretation as being those hypotheses which the sensor is best able to support.

The model has the general parameterisation

$$P(Y, D, W) = P(W) \prod_{i=1}^{n} \{P(y_i|d_i, W) P(d_i|W)\}, \tag{2.14}$$

from which the authors make simplifying independence assumptions to give

$$P(Y, D, W) = P(W) \prod_{i=1}^{n} \{P(y_i|d_i)\ P(d_i|W)\}. \tag{2.15}$$

As noted by the authors, if the sensor $x_i$ is an optical sensor such as the Landsat TM or SPOT satellite, then $P(y_i|d_i)$ can be specified by multivariate Gaussian distributions.

In their model, $W$ is a set of hypotheses for which sensor $x_i$ provides evidence on $W$ via the classes $d_i$.

In the above, Bayes rule and marginalisation are used to manipulate the distribution to obtain the posterior distribution for $W$, that is $P(W|x_1 = a, x_2 = b, \ldots, x_m = z)$, where $a, b, \ldots z$ are the data values for some pixel.

A limitation of the above model is its simplistic consideration of independence assumptions; all data sources are assumed to be independent given $W$.

Further, the manipulation of Bayes rule and the calculations required for marginalisation may change, requiring specific implementations. For this reason, a generic approach appears advantageous. This is reviewed in section 2.3.3, and in section 2.3.4 a graphical means for constructing probabilistic knowledge–based systems is described.

### 2.3.2 Basic Concepts of Dempster–Shafer Theory of Evidence

The DS theory of evidence aims to model and quantify uncertainty by *degrees of belief*. In contrast to Bayesian approaches, the theory allows *probability masses* to be assigned to sets of hypotheses rather than hypotheses in isolation. In this way,

information sources possibly operating at different levels of generality are used to narrow down the hypothesis set.

Various evidence theory models have been used in the RS/GIS fields for combining data and for reasoning with uncertainty. Perhaps the most popular models are those based upon the classical approach to evidence theory as proposed by Shafer (1976). His model is essentially based on the notion of belief functions and Dempster's rule of combination. This view does *not* rely on probability theory and is ofter called the transferable belief model.

Evidence theory provides a framework for combining data received from multiple sensors (experts) for the purpose of interpreting the data. Generally the interest is in making a decision from a set of $n$ mutually exclusive, competing hypotheses $\Omega$:$\{\omega_i | i = 1, \ldots, n\}$.

Next consider a set of sensors $\Theta$ that provide information on $\Omega$ via a (typically unobserved) mapping $\Gamma = \Theta \rightarrow 2^{\Omega}$. Note that the mapping need not map a sensors output to the atomic hypothesis $\omega_i$ , reflecting the belief that some sensors are unable to resolve between certain hypotheses (or have no concept of two outcomes being distinct) and thus can only provide information on sets of hypotheses. Further, each sensor will have an uncertainty associated with its measurements and hence the information it provides.

The information obtained from each sensor is represented as a *mass* distribution, defined as follows:

each function $m : 2^{\Omega} \rightarrow [0, 1]$ is called a mass distribution whenever

$$m(\emptyset) = 0,$$

and

$$\sum_{A:A \subseteq \Omega} m(A) = 1, \tag{2.16}$$

where $\emptyset$ denotes the empty set.

The quantity $m(A)$ is understood to be the measure of belief given to $A$ but

not to any strict subset of $A$. Those sets $A$ with $m(A) > 0$ are called *focal sets.*

A (possibly unnormalised) mass distribution defined on $\Omega^A$ will be denoted $\psi^A$.

To interpret a body of evidence relative to a question, $A$, the evidential interval $[Bel(A), Pl(A)] \subseteq [0,1]$ is calculated as follows:

$$Bel; 2^\Omega \to [0,1]; \quad Bel(A) = \sum_{B:B\subseteq A} m(B)$$
$$Pl; 2^\Omega \to [0,1]; \quad Pl(A) = \sum_{B:A\cap B\neq\emptyset} m(B) = 1 - Bel(\bar{A})$$

The value $Bel(A)$ measures the *total belief* (or *support*) for $A$, that is the sum of the mass associated with $A$ and all its subsets. The quantity $Pl(A)$ measures the degree to which the evidence fails to refute $B$, or in other words the degree to which $A$ remains *plausible.* The interpretation of $Pl(A)$ as measuring *remaining plausibility* comes from the observation that the conditioning scheme used in this theory can *relocate* mass from $A$ to $B$ if $A \cap B \neq \emptyset$. Sometimes encountered in the literature is the interpretation of the evidential interval as a bound on the true probability of an event, that is $Bel(A) \leq P(A) \leq Pl(A)$, but this interpretation is not correct as the belief and plausibility functions are not strictly probabilities and there is no guarantee that the relationship holds.

One of the main characteristics of the DS approach is that it presents a formalism for combining distinct pieces of evidence. This rule is often called *Dempster's Rule of Combination.* Given two basic belief masses $m_1$ and $m_2$ pertaining to two sources of evidence, then the result of the combination is a new belief mass $m_1 \oplus m_2$ defined by

$$(m_1 \oplus m_2)(C) = \frac{\sum_{A,B:A\cap B=C} m_1(A)m_2(B)}{\sum_{A,B:A\cap B\neq\emptyset} m_1(A)m_2(B)}, \quad C \neq \emptyset, \quad C \subseteq \Omega,$$

and

$$(m_1 \oplus m_2)(\emptyset) = 0.$$

The distribution $m_1 \oplus m_2$ exists if and only if

$$K = \sum_{A,B:A\cap B=\emptyset} m_1(A)m_2(B) < 1. \tag{2.17}$$

The operator $\oplus$ is commutative and associative and hence the final result does not depend on the order in which the information is introduced and combined. The operator $\oplus$ treats the distributions $m_1$ and $m_2$ as though they are independent.

The quantity $K$ of equation 2.17 reflects the partially conflicting situation where two sets $A$ and $B$ have an empty intersection but both $m_1(A)$ and $m_2(B)$ are different from zero, which would imply that mass should be attributed to the empty set and be in conflict with definition 2.16. Dempster's rule solves this problem by forcing $(m_1 \oplus m_2)(\emptyset) = 0$ and renormalising. When two information sources produce completely conflicting statements, then $K = 1$ and $m_1 \oplus m_2$ is not defined.

Dempster's rule of combination can combine different sources of evidence when they are expressed relative to a common frame of discernment. When evidence expressed relative to different frames needs to be combined, then the evidence (mass) needs to be *projected* to a common frame so that Dempster's rule can be applied.

Consider two frames $\Omega$ and $\Omega^*$, where the variables defined in $\Omega$ are a subset of those defined in $\Omega^*$. To project mass from one frame to the other requires the mappings

$$\hat{\prod} : 2^{\Omega} \rightarrow 2^{\Omega^*} \tag{2.18}$$

and

$$\prod : 2^{\Omega^*} \rightarrow 2^{\Omega} \tag{2.19}$$

be defined. Equation 2.18 must satisfy the following conditions:

1. No focal elements can map to the empty set.

2. Two mutually exclusive elemental hypotheses must remain mutually exclusive under the mapping.

3. Elements in $\Omega^*$ must originate from elements in $\Omega$.

4. The projection of a proposition must equal the union of the projection of its constituent elements.

Conditions on mapping 2.19 follow from the above four conditions.

The mapping $\hat{\prod}$ is called a *refinement mapping*, while the set $\Omega^*$ is called a *refinement* of $\Omega$ as it is able to carry more information than $\Omega$. In this context, $\Omega$ is said to be a *coarsening* of $\Omega^*$.

Given the above two mappings, and mass distributions $m(A)$ and $m^*(A^*)$ defined on $\Omega$ and $\Omega^*$ respectively, mass is projected between the two frames via the operations

$$m(A) = \sum_{A^* \subseteq \Omega^*; \hat{\prod}(A^*)=A} m^*(A^*); A \subseteq \Omega, \tag{2.20}$$

and

$$m^*(A^*) = m(A) \quad \text{if } A^* = \hat{\prod}(A), \quad 0 \text{ otherwise.} \tag{2.21}$$

When $A$ is a singleton hypothesis, that is $A = \omega_i; \omega_i \in \Omega$, the mass distributions are probability distributions and hence the above formulae can be used to project probabilities.

In the remainder of this thesis, the notation $\psi^{X \to Y}$ will be used to represent the projection of mass from $\Omega^X$ to $\Omega^Y$, where is is understood that this may involve the application of equation 2.20 or equation 2.21 or both.

### Evidence Theory Models in Remote Sensing, Geographical Information Systems and Artificial Intelligence

Evidential reasoning has been suggested as a methodology for the *multisource data combination problem* (Lee et al., 1987) and applied in practice to different

problems including land use classification (Wilkinson and Mégier, 1990), helicopter route planning (Garvey, 1987) and the interpretation of geological data (Moon, 1990).

In evidence theory, each data source provides its evidence in the form of a mass distribution defined over some frame of discernment, for example the frame of discernment defined by the set of competing land use classes (Wilkinson and Mégier, 1990). Generally, data are specified as a mass distribution, typically where the weights of the mass function (or rules) are defined by a (meta) expert. When remotely sensed data have been interpreted by a MLC, the probability of class membership may be used (Wilkinson and Mégier, 1990).

Mass distributions may be derived from ancillary data by considering a human expert's opinion of how a unit of mass should be distributed across the frame of discernment. Prior knowledge is often combined in the model by considering it as another information source. When multiple sources of evidence are available, then equation 2.17 is used to combine the evidence and a decision of land use is then made.

The following (ficticious) example illustrates the use of equation 2.17 for combing evidence obtained from multiple sources. In this case, one of the sources is the output of a MLC and the other source is evidence resulting from expert knowledge.

Suppose we wish to produce a land use map of an agricultural region, where the land use units are wheat ($W$), lupins ($L$), pasture ($P$) and remnant vegetation ($R$). Further assume there are two sources of information: the output of a MLC and a human expert.

The first step of the problem is to define the frame of discernment, which in this case is simply $\Omega^C = \{W, L, P, R\}$.

The second step is to define how the information sources relate to this frame and form a mass distribution for each information source.

Let $m_C^{rs}$ be the mass distribution constructed by assigning to each elemental hypothesis the posterior probabilities obtained from the maximum likelihood clas-

sifier and let $m_C^{ex}$ be the mass distribution constructed from consideration of the expert knowledge.

Assume that for a particular pixel $m_C^{rs}(W) = 0.5$, $m_C^{rs}(L) = 0.3$, $m_C^{rs}(P) = 0.1$, and $m_C^{rs}(R) = 0.1$. Assume further that we have an estimate that the classification has an accuracy of 90%. In evidence theory models, estimates of ignorance of information sources are often accomodated by assigning some mass to the set $\Omega^C$, in the same way that mass is attributed to other sets of hypotheses when we are uncertain about how to divide it among subclasses, or unable to divide it. The 90% accuracy estimate (10% ignorance) is used to modify $m_C^{rs}$ by assigning 0.1 units of mass to $\Omega^C$ and then multiplying each of the other masses by 0.9. The modified (unnormalised) mass distribution is $\tilde{m}_C^{rs}(W) = 0.45$, $\tilde{m}_C^{rs}(L) = 0.27$, $\tilde{m}_C^{rs}(P) = 0.09$, $\tilde{m}_C^{rs}(R) = 0.09$ and $\tilde{m}_C^{rs}(\Omega^C) = 0.1$.

Assume that $m_C^{ex}$ was similarly discounted to obtain the distribution $\tilde{m}_C^{ex}$, defined as follows: $\tilde{m}_C^{ex}(\Omega^C) = 0.4$ and $\tilde{m}_C^{ex}(\bar{P}) = 0.6$, where $\bar{P}$ represents the complement of $P$ (that is, *not* pasture).

Applying Dempster's rule of combination, equation 2.17, the joint mass for wheat $(W)$ is

$$(\tilde{m}_C^{rs} \oplus \tilde{m}_C^{ex})(\{W\}) = N^{-1}(\tilde{m}_C^{ex}(\bar{P}) \times \tilde{m}_C^{rs}(\{W\}) + (\tilde{m}_C^{ex}(\Omega^C) \times \tilde{m}_C^{rs}(\{W\})) \quad (2.22)$$

where $N^{-1}$ is the normalising constant. The masses for the other classes are similarly calculated from equation 2.17 to obtain

$$\begin{aligned}
(\tilde{m}_C{}^{rs} \oplus \tilde{m}_C^{ex})(\Omega^C) &= 0.042, \\
(\tilde{m}_C{}^{rs} \oplus \tilde{m}_C^{ex})(\{W\}) &= 0.476, \\
(\tilde{m}_C{}^{rs} \oplus \tilde{m}_C^{ex})(\{L\}) &= 0.285, \\
(\tilde{m}_C{}^{rs} \oplus \tilde{m}_C^{ex})(\{P\}) &= 0.038, \\
(\tilde{m}_C{}^{rs} \oplus \tilde{m}_C^{ex})(\{R\}) &= 0.095 \text{ and} \\
(\tilde{m}_C{}^{rs} \oplus \tilde{m}_C^{ex})(\bar{P}) &= 0.063.
\end{aligned}$$

The belief and plausibility ranges $[Bel(A), Pl(A)]$ are:

$$\begin{aligned}
\Omega^C &: [1.0, 1.0], \\
\{W\} &: [0.476, 0.582], \\
\{L\} &: [0.285, 0.391], \\
\{P\} &: [0.038, 0.080], \\
\{R\} &: [0.095, 0.201], \\
\bar{P} &: [0.920, 0.962].
\end{aligned}$$

Using the decision rule that the elemental proposition with the highest mass is chosen as the result, this pixel would be given label $W$.

The above example highlights the following:

1. Data source reliabilities can be modelled by assigning a proportion of mass to $\Omega^C$.

2. Masses can be assigned to arbitrary sets of hypotheses, which allows flexibility and ease in the expression of masses.

3. After combining masses, $\tilde{m}_C(\Omega^C)$ can be interpreted as a measure of ignorance (Wilkinson and Mégier, 1990). If it is greater than the mass of all other singleton hypotheses, then choosing an hypothesis that is believed less than the expression of ignorance would seem unnatural. In this case, Wilkinson and Mégier (1990) suggest (in the context of hierarchical hypothesis spaces) choosing a superclass that has maximum belief greater than the level of ignorance as the pixel label.

4. Belief and plausibility values give a second-order indication as to the quality of the results. If the evidential intervals of a number of hypotheses overlap,

then choosing one of these hypotheses in preference to the others contains an element of risk, as the unchosen hypotheses to a large degree still remain unrefuted. Incorporating the intervals into the interpretation (decision) process is examined below.

The above example illustrates the basic mechanisms of combining evidence using evidence theory. Aspects of decision rules are now discussed.

### Decision Rules

The above example used the decision rule that the singleton hypothesis possessing the largest resulting mass is chosen as the pixel label. This rule did not, however, take into account the evidential intervals available for each proposition.

Lee et al. (1987) suggest using the evidential interval $[Bel(A), Pl(A)]$ to place a decision in the context of *risky* or *confident*. According to Lee et al. (1987), a decision may be considered risky if the difference in beliefs of the two most likely labels is smaller than the evidential interval of the most favoured label. The idea behind this is that the *probability* of the most favoured label $A$, say, falls somewhere in the interval [Bel(A),Pl(A)] and the *probability* of the second most favoured label may also fall in this region. A decision may be considered confident if the difference in the supports is greater than the evidential interval. The motivation of Lee et al. (1987) is that quality assessment of the decision appears to be driven by the notion of overlapping intervals, although they do not explicitly consider the overlap. Perhaps a better rule would be formed by considering the intervals of the two most favoured labels and grading the degree of risk based upon the degree of overlap.

Another decision problem arises when the masses of the elemental hypotheses are all less than the mass of $\Omega^C$. If the hypothesis space is hierarchically ordered, then one approach is to choose the superclass with the highest belief, where this belief must be greater than the level of ignorance. This approach was used by Wilkinson and Mégier (1990) to assign superclasses such as *cereal grains* if none

of the elemental classes such as wheat and barley could be assigned.

The above example is a simple illustration of combining data using the DS theory of evidence. In the above example, data and other evidence are presented as mass distributions which are combined using Dempster's Rule. Methods to be presented in the following section illustrate how calculations in an arbitrary domain are performed, and thus how relatively complex domains may be defined.

### 2.3.3 Exploiting Domain Structure — Propagating Beliefs in Hypertrees

Updating beliefs in probability and DS theory is in general computationally expensive. In the case of probabilities, calculations are performed on the product space of the variables. The size of the product space may be relatively large even for a small set of variables. For example ten variables each having five states have a product space with 9765625 entries for which computations need to be performed. Computations for the DS approach include those for the probabilistic case plus computations associated with all possible subsets of the variables.

Computationally efficient combination schemes/algorithms may be devised by enforcing or exploiting the structure of a domain. This section examines how domain structure may be exploited in the data combination process.

Computational schemes that exploit or impose structure have been devised for both probability (Pearl, 1986b; Lauritzen and Spiegelhalter, 1988; Shenoy and Shafer, 1986, 1990) and DS-based (Shenoy and Shafer, 1986, 1990) models. In their most general forms, such schemes rely on the structure of the problem being formalised as a graph which has some special properties which allow local computations to be used.

Two conditions must be met so that local computations may be exploited:

1. the joint probability distribution or the joint belief function must factor into functions each involving a small set of variables; and

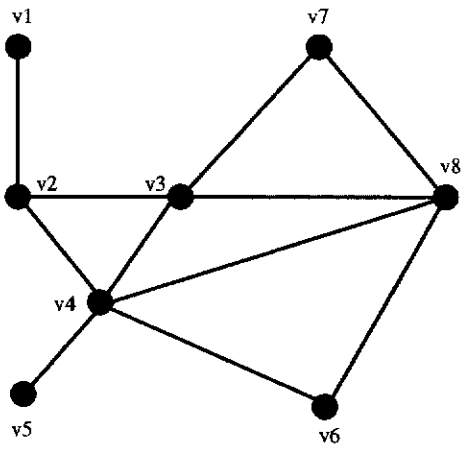



Figure 2.1: A graphical depiction of an undirected graph.

2. these sets must form a hypertree.

Such special properties relate to the structure of the domain, expressed in terms of graph theory, which is introduced below.

### Graphs, Hypergraphs and Hypertrees

A *graph* $G$ is a pair $(V, E)$ where

1. $V = \{v_1, \ldots, v_i\}$ is a set of elements called *vertices*, and

2. $E = \{e_1, \ldots, e_k\}$ is a set of pairs of vertices $e_k = (v_i, v_j),\ v_i, v_j \in V$, called *edges*. The vertices $v_i$ and $v_j$ are said to be *neighbours*.

In the following, an edge $(v_i, v_j)$ will be denoted $e_{i,j}$, or simply $e$ where it is clear which edge is being referred to. A vertex of a graph is called a *leaf* if it has only one neighbour. An edge $e = (v_i, v_j)$ is said to be *directed* from $v_i$ to $v_j$ if the ordering of the vertex pair is important, otherwise the edge is said to be *undirected*. A graph constructed from directed edges is called a *directed graph* and a graph constructed from undirected edges is called an *undirected graph*.

An example of an undirected graph is given in figure 2.1. For this graph, $V = \{v_1, v_2, \ldots, v_8\}$ and $E = \{e_{1,2}, e_{2,3}, e_{3,4}, e_{2,4}, e_{4,5}, e_{3,7}, e_{3,8}, e_{4,8}, e_{4,6}, e_{6,8}, e_{7,8}\}$.

In a directed graph, vertex $v_j$ is called a *child* of $v_i$ if the edge $e_{i,j} = (v_i, v_j)$ exists, and $v_i$ is then called a *parent* of $v_j$. For a given vertex $\alpha$, say, let $pa(\alpha)$ and $ch(\alpha)$ represent the set of all vertices that are parents and children of $\alpha$ respectively.

If $v_1, v_2, \ldots, v_p$ is a sequence of distinct vertices, where $p > 1$, and $(v_k, v_{k+1}) \in E$ for $k = 1, 2, \ldots, p-1$, then $v_1, v_2, \ldots, v_p$ is called a *path* from $v_1$ to $v_p$, and denoted $p_{1,p}$.

A path is considered to be either directed or undirected, depending upon whether it is composed of directed or undirected edges respectively. A path is said to be a *simple path* if $v_i \neq v_j$ for $i \neq j$ for $1 \leq i < j \leq p$.

A cycle of length $p$ is a path $\{v_1, v_2 \ldots v_p, v_1\}$, $v_i \in V$ for $i = 1, \ldots, p$, from a vertex $v_1$ to $v_1$. A cycle is said to be a *simple cycle* if $\{v_1, v_2 \ldots v_p\}$ is a simple path.

In an undirected graph, a simple cycle posseses a *chord* if there exists an edge between two nonconsecutive vertices of the cycle. If a cycle does not possess a chord it is said to be *chordless.*

A *connected* graph, where edge directions are ignored, contains a path $p_{i,j}$ for each pair of distinct vertices. In the following, it is assumed that all graphs are connected.

A *tree* is a connected graph such that, if the edges are taken to be undirected, it has no cycles.

A directed graph that contains no directed cycles is called a *directed acyclic graph* or a DAG for short.

Efficient probability and DS propagation schemes exist for graphs having the property that they can be transformed into a *hypergraph*, which has certain special properties.

Let $\Delta = (E_i | i \in I)$ be a family of subsets of $V$. The family $\Delta$ is said to be a *hypergraph on* $V$ if

1. $E_i \neq \emptyset$, $i \in I$, and

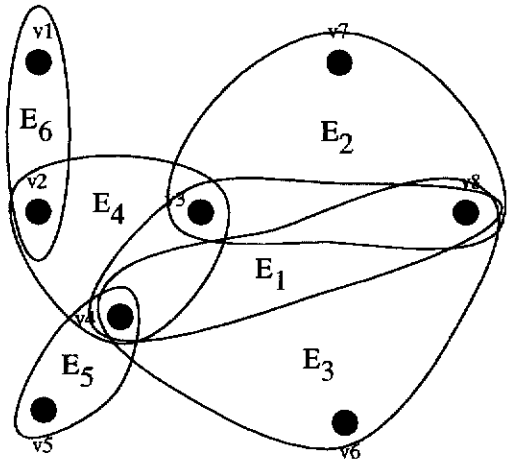



Figure 2.2: A graphical depiction of a hypergraph.

2. $\bigcup_{i \in I} E_i = V$.

Then the pair $H = (V, \Delta)$ is called a *hypergraph* and the elements of $\Delta$ are called the *hyperedges*. The two conditions above simply state that in order to be a hypergraph, a hyperedge must contain at least one vertex of the underlying graph and each vertex in the underlying graph must be included in at least one hyperedge.

A hypergraph representation of figure 2.1 is depicted in figure 2.2, where the closed curves enclosing vertices are the hyperedges

$$\Delta = \{\{v_1, v_2\}, \{v_2, v_3, v_4\}, \{v_3, v_7, v_8\}, \{v_3, v_4, v_8\}, \{v_4, v_8, v_6\}, \{v_4, v_5\}\}.$$

Suppose $E_a$ and $E_b$ are distinct hyperedges in a hypergraph $H$, $E_a \cap E_b \neq \emptyset$, and $E_b$ contains every vertex of $E_a$ that is contained in a hyperedge of $H$ other than $E_b$. Then $E_a$ is called a *twig* of $H$ and $E_b$ a *branch* for $E_a$. For example, in the above diagram, if $E_2$ was considered a twig, then $E_1$ is a possible branch for $E_2$.

A hypergraph is called a *hypertree* if there is an ordering of its hyperedges, say $h_1, h_2, \ldots, h_k$, such that $h_k$ is a twig in the hypergraph $\{h_1, h_2, \ldots, h_k\}$ whenever $2 \leq k \leq n$; $n = |\Delta|$, the number of hyperedges in $H$. Any such ordering of the hyperedges is called a *hypertree construction sequence* for the hypertree. The first

hyperedge in a hypertree construction sequence is called the *root* of the hypertree construction sequence.

From figure 2.2, if $E_1$ is the root, then the hyperedges considered in the order $E_1, E_2, E_3, E_4, E_5, E_6$ is a possible hypertree construction sequence.

If a hypertree is constructed by adding hyperedges following a hypertree construction sequence, then each added hyperedge is a twig and has at least one branch in the hypertree at that point. Suppose such a branch, say $\beta(h)$, is chosen for each hyperedge $h$ added. By doing so, a mapping $\beta$ from $H\backslash\{h\}$ to $H$ is constructed. This mapping is called a *branching* for the hypertree construction sequence.

A twig may have more than one branch and hence a hypertree may have more than one branching. For each hyperedge of a hypertree, there is at least one construction sequence beginning with that hyperedge.

If the sets of variables form a hypergraph which is not a hypertree, then the hypergraph must be transformed into a hypertree. This can be done by replacing existing hyperedges with larger hyperedges. Replacing hyperedges by larger hyperedges amounts to adding edges to the underlying graph and is often referred to as *filling-in* (the graph), while the edges added are referred to as *fillin* edges. A class of graphs called *triangulated* or *chordal* graphs has the required property of being able to be represented as a hypertree. Triangulation of graphs is considered in section 4.3.

If a hypergraph is not a hypertree, then it is generally made so by triangulating the underlying graph. In the worst case, such a transformation will result in a hyperedge that consists of all vertices of the underlying graph, and no computational advantage will be achieved. However, (probabilistic) independence (or as defined by Pearl (1986b), *qualitative conditional independence* for DS mass distributions) assumptions are typically specified to avoid this difficulty.

The joint distribution of a number of variables may be depicted as a graph, where a vertex is associated with each variable and an edge exists where two

variables are jointly considered. In this way, the structure of the joint distribution may be formalised by considering the structure of the graphical representation.

The main purpose of the hypertree and its construction sequence is that it provides an efficient means of ordering probability and belief calculations used when revising the joint distribution; that is, when updating beliefs.

### Combining Evidence Using Hypergraphs

This section provides an overview of the work of Shenoy and Shafer (1990), although here the propagation scheme proposed by Lauritzen and Spiegelhalter (1988) is explicitly described using the terminology used by Shenoy and Shafer (1990). In this way, comparison of the different schemes is more evident.

In the following discusion, it is assumed that the joint belief function is graphically represented as a hypertree $T$, and is parameterised by a set of functions called *valuations*. Valuations are unnormalised probabilites or mass distributions, depending upon whether probabilities or DS mass functions are being considered respectively.

The computation schemes rely on a partitioning of the variables such that the partitioning can be represented by a hypertree $T = (V, \Delta)$, say, and the joint distribution $\psi^V$ can be parameterised into a set of functions $\psi^N, N \in \Delta$, called *valuation functions*. Computational efficiency is achieved by performing calculations local to the sets of variables $N \in \Delta$.

Consider a product space $\Omega^V$. The valuation functions attach non-negative real numbers to non-empty subsets and are defined as

$$\Psi = \{\psi^N : 2^{\Omega^N} \rightarrow \Re_0^+ | N \subseteq V\} \tag{2.23}$$

Let $\Psi_N$ denote the valuations on $\Omega^N$. Depending upon the type of valuation that is actually considered (eg. probabilities or basic belief assignments), only a subclass $A_N \subseteq \Psi_N$ of valuations allows the application of the local computation scheme. The elements $A_N$ are called *admissible* valuations.

In the case of probabilities, valuations correspond to unnormalised marginal probabilities, which are often referred to as *potentials* (Lauritzen and Spiegelhalter, 1988). In the case of DS theory, valuations correspond to unnormalised mass distributions.

The final aim of the approach is to calculate *marginals* of the overall valuation $\psi^V$ which arises from the *aggregation* of partial valuations $\psi^N, N \subseteq V$. For this purpose, two main operations, *combination* and *marginalisation*, are applied.

Assume there is a mapping $\otimes : \Psi \times \Psi \rightarrow \Psi$, called *combination* such that:

1. If $\psi^G$ and $\psi^H$ are valuations on $\Omega^G$ and $\Omega^H$ respectively, then $\psi^G \otimes \psi^H$ is a valuation on $\Omega^{G \cup H}$.

2. If either $\psi^G$ or $\psi^H$ is not a proper valuation, then $\psi^G \otimes \psi^H$ is not a proper valuation.

3. If $\psi^G$ and $\psi^H$ are proper valuations, then $\psi^G \otimes \psi^H$ may or may not be a proper valuation.

If $\psi^G \otimes \psi^H$ is not a proper valuation, then we say that $\psi^G$ and $\psi^H$ are *not combinable*. If $\psi^G \otimes \psi^H$ is a proper valuation, then $\psi^G$ and $\psi^H$ are *combinable*, and $\psi^G \otimes \psi^H$ is the *combination of* $\psi^G$ *and* $\psi^H$.

Intuitively, combination corresponds to aggregation. If $\psi^G$ and $\psi^H$ represent information about variables in $\Omega^G$ and $\Omega^H$ respectively, then $\psi^G \otimes \psi^H$ represents the aggregated information for the variables in $\Omega^{G \cup H}$. In the probability case, combination corresponds to pointwise multiplication (see equation 2.13). In the belief function case, combination uses Dempster's Rule (see equation 2.17).

In the case of probabilities, non-combinable valuations result in definition 2 not being satisfied for some subspace, and in the case of DS, equation 2.17 not being satisfied.

Given a hypertree $T = (V, \Delta)$ and a valuation $\psi^N \in A_N$ specified for each hyperedge $N \in \Delta$, the process of combining data is:

1. to check whether the valuations $\psi^N$ are combinable,

2. to carry out the aggregation, and finally

3. to project the aggregated valuation to the subspace of interest.

That is, the aim is to calculate, for each hyperedge $N \in \Delta$, the quantity

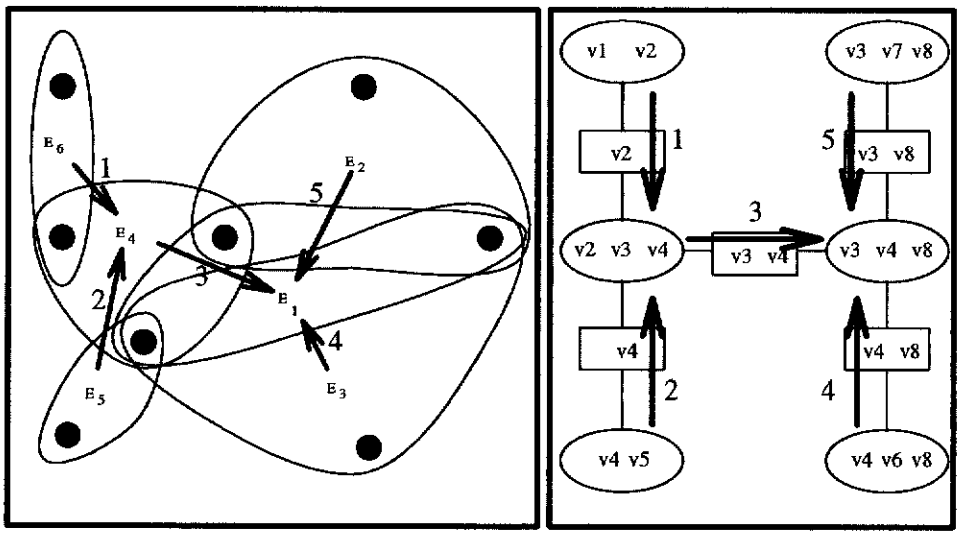
$$\psi^N = (\otimes_{S \in \Delta} \psi^S)^{S \to N}. \tag{2.24}$$

Without exploiting structure, this generally would not be feasible. For example, given 30 binary variables, combination using probability theory and DS theory would in general require $2^{30}$ and $2^{2^{30}}$ values respectively to be calculated.

Consider a hypertree $T$ where evidence has been entered. The first step in updating beliefs is to condition a valuation (which corresponds to equation 2.13 for probabilities and equation 2.17 for DS belief functions) for which the variable (or set of variables) is defined, $\psi^A; A \in \Delta$ say, to obtain the (revised) valuation $\psi_c^A$. Assume that all evidence has been entered to give the revised valuations $\psi_c^A; \forall A \in \Delta$, where $\psi_c^A = \psi^A$ if no evidence directly relating to $A$ is entered. The next step is to project the updated beliefs to the subspace of interest. The construction sequence for $T$ provides an ordering in which to perform the calculations. That is, given a construction sequence $H = \{h_1, h_2, \ldots, h_n\}$, $n = |\Delta|$, recursively perform the following calculations until the marginal (revised) valuation is obtained (up until normalisation) for the tree root:

$$\text{for } i = n \ldots 2 \;\; \psi_c^{\beta(h_i)} := \psi_c^{\beta(h_i)} \otimes (\psi_c^{h_i \to \beta(h_i)}). \tag{2.25}$$

The process of projecting quantities from one set of variables to another is often considered as *message passing*, where one set of variables passes a message (the projected quantity) to another set of variables. This analogy arises from considering distributed versions (Shenoy and Shafer, 1990) of the algorithm, where messages need to be physically communicated amongst a number of processors.




(a) (b)

Figure 2.3: Graphical depiction of message-passing using construction sequence (a) hypertree (b) Markov tree representation.

Considering figure 2.2 and the construction sequence $H = \{E_1, E_2, E_3, E_4, E_5, E_6\}$, the flow of messages is graphically depicted in figure 2.3(a), where the arrows represent messages sent and are numbered in the order of their execution.

The above calculation updates our belief for the variables in one hyperedge. Clearly, updating the beliefs for all variables could be achieved by consecutively considering each hyperedge as the root from which a hypertree construction sequence is generated and then performing equation 2.25. This, however, would result in much duplication of computational effort.

The probability propagation algorithm proposed by Lauritzen and Spiegelhalter (1988) uses the following propagation strategy to obtain revised marginals for all hyperedges:

$$\begin{aligned}
\text{for } i = n, n-1, \ldots, 2 \quad & \psi_c^{\beta(h_i)} := \psi_c^{\beta(h_i)} \otimes (\psi_c^{h_i \to \beta(h_i)}) \\
& \psi_c^{h_i} := \psi_c^{h_i} \otimes \frac{1}{\psi_c^{\beta(h_i) \to \beta(h_i) \cap h_i}} \\
\text{Normalise} \quad & \psi_c^{h_1} \\
\text{for } i = 2, 3, \ldots, n \quad & \psi_c^{h_i} := \psi_c^{h_i} \otimes \psi^{\beta(h_i) \to h_i}
\end{aligned} \tag{2.26}$$

The above scheme is intuitively appealing as the steps have the following interpretation:

- The first step reparameterises each clique $h_i$ to have distribution $P(R_i|S_i)$ where $R_i, S_i \subset V$, $S_i = \beta(h_i) \cap h_i$, and $R_i = h_i \backslash S_i$ for $i = 2, 3, \ldots, n$.
- The second step forms the distribution $P(S_1) = P(h_1)$, from which by the chaining rule we obtain $P(S_1) \prod_{i=2,\ldots,n} P(R_i|S_i)$.
- The revised marginals $P(R_i, S_i) = P(R_i|S_i)\ P(S_i)$, $i = 2, 3, \ldots, n$ are calculated.

The scheme of Shenoy and Shafer (1990) uses a slightly different approach, and propagates both probabilities and DS belief masses with the added advantage of allowing distributed versions of the algorithm to be implemented. In this scheme, calculations are further structured to avoid computational duplication.

To explain this further structuring, it is convenient to use a *Markov Tree* representation of the hypertree. A Markov tree representation $M = (P, Q)$, for a hypertree $T = (V, \Delta)$, is a graph where vertex $p_i = E_i$, $p_i \in P$, $E_i \in \Delta$ and an edge $q = (p_x, p_y)$, $q \in Q$, is defined if $E_x \cap E_y \neq \emptyset$. For this representation, each vertex $p_i \in P$ can be considered as a processor which is able to communicate with its neighbouring processors, $p_j \in P$, via a communication link $(p_i, p_j) \in Q$.

Figure 2.3(b) shows the Markov tree representation of figure 2.2, where ovals represent the Markov tree vertices and the edges are represented as rectangles joining the vertices. The labels of the edges represent the sets of vertices that the neighbouring (as defined by that edge) vertices have in common. The bold arrows represent the flow of messages and are analogous to those in figure 2.3(a). Updating beliefs for all variables is then achieved by a message-passing scheme, which is just an efficient way of performing equation 2.25 for each set of variables.

A vertex $v$ sends to its neighbours $u \in Ne(v)$ the message

$$\psi_*^{v \to u} = (\psi_c^v \otimes \{\otimes \psi_*^{s \to v} | s \in Ne(v) \backslash u\})^{v \to u} \tag{2.27}$$

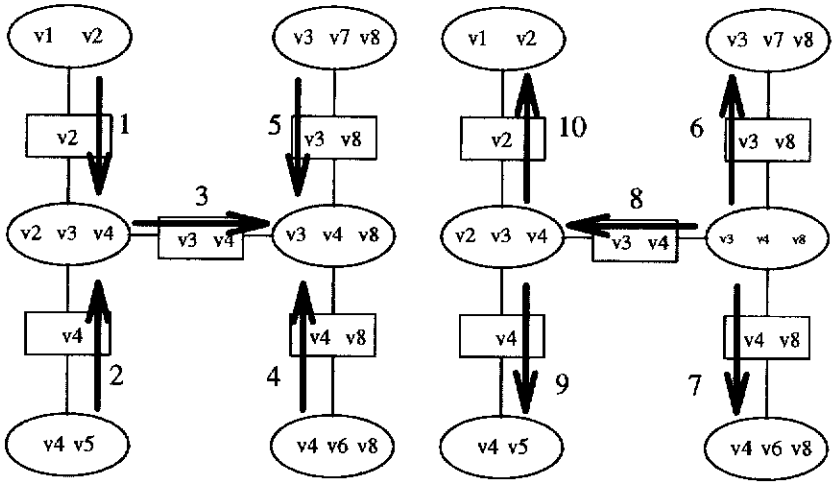



Figure 2.4: Graphical depiction of message passing using a Markov tree representation.

The message is only calculated and sent once $v$ has received all the messages from its neighbours $Ne(v)\backslash u$. Note that messages $\psi_*$ are projected and *stored* locally by the processor until the processor is ready to combine the messages.

When $v$ has received messages from all $Ne(v)$, it then evaluates its marginal by calculating

$$\psi^v_{new} = \psi^v_c \otimes \{\otimes \psi^{r\to v}_* | r \in Ne(v)\}. \tag{2.28}$$

The calculation for updating beliefs can be instigated by instructing all leaves to send their messages. The flow of messages is then from the leaves to the middle vertex(s), and then from the middle vertex(s) back out towards the leaves.

On a single-processor machine, the above calculations can be ordered by making use of a hypertree construction sequence, where the construction sequence applies to the appropriate Markov tree vertex, which was derived from $T$. That is:

1. For $i = n \ldots 2$, send the message $\psi^{h_i\to pa(h_i)}_* = \psi^{h_i}_c \otimes \{\otimes \psi^{r\to h_i}_* | r \in ch(h_i)\}$.

2. For $i = 1 \ldots n-1$, send the message

$$\psi^{pa(h_i)\to h_i}_* = \psi^{pa(h_i)}_c \otimes \{\otimes \psi^{s\to pa(h_i)}_* | s \in ch(h_i)\backslash s\}.$$

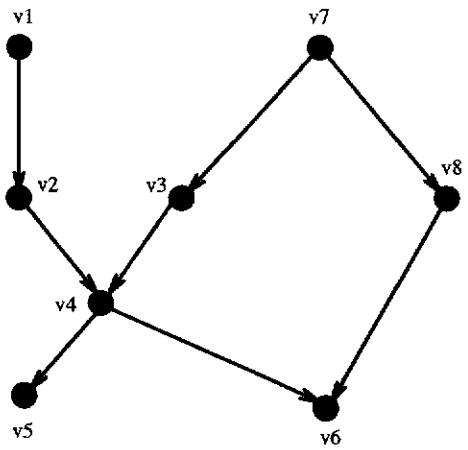



Figure 2.5: Graphical depiction of a directed acyclic graph (DAG).

3. $\forall\, v \in P$, calculate the marginal from equation 2.28.

The flow of messages is depicted in figure 2.4.

### 2.3.4 Bayesian Networks — A Graphical Tool

A Bayesian network, also referred to as a *causal network*, a *conditional probability network* or simply a (Bayesian) *belief network*, can be represented by a DAG, where the vertices of the graph represent random variables (often referred to as pieces of evidence or hypotheses in the AI literature) and the edges of the graph represent relationships that exist between the variables. Vertices that represent variables for which we have direct observations (i.e. data or *evidence*) are called *observed* vertices. Vertices for which no direct observations are available, but whose states are inferred from evidence applied at observed vertices, are called *unobserved* vertices and the corresponding variables *unobserved* variables. Variables may be *continuous* or *discrete* (Olesen, 1993).

The knowledge embodied in the network is represented by the joint probability distribution of the random variables, which is parameterised by applying the *chaining rule* of equation 2.8. For example, the DAG shown in figure 2.5 has the following joint distribution:

$$P(v_1, v_2, \ldots v_8) = P(v_6|v_4, v_8)P(v_8|v_7)P(v_5|v_4)P(v_4|v_2, v_3)P(v_2|v_1)P(v_7)P(v_1) \tag{2.29}$$

As is apparent, rules relating to a variable $v$ must be expressed relative to the set of parent variables $pa(v)$. For example, $P(A|B, C)$ represents the set of rules: *if* $B = b_i$ and $C = c_j$, then $A = a_k$ with probability $P(A = a_k|B = b_i, C = c_j)$.

Given a joint distribution such as that of equation 2.29, the interest is typically in obtaining revised beliefs given some new piece of evidence. For example, if the evidence $v_1 = k$ is observed, then the network belief is updated to $P(v_2, v_3, \ldots, v_8|v_1 = k)$ using the calculations presented in section 2.3.3, which duplicate that which would be achieved by applying Bayes rule, equation 2.12.

To make use of the computation scheme of section 2.3.3, a hypertree representation of the DAG needs to be made. Assuming a DAG $D = (V, E)$, this is achieved by the following steps:

1. Marry the parents and drop the edge directions, that is: $\forall v \in V$ and $\forall p, q \in pa(v)$, $E = E \cup (p, q)$.

2. Triangulate the graph (if it is not already triangulated) with an algorithm such as that described by Kjaerulff (1990).

3. Extract all maximal cliques using an algorithm such as that described by Bron and Kerbosch (1973).

4. Form a branching for the cliques using maximum cardinality search (Tarjan and Yannakakis, 1984) as described by Lauritzen and Spiegelhalter (1988).

5. From the branching sequence, form a Markov tree representation for $D$. Figure 2.4 is a possible representation for the DAG depicted in figure 2.5.

6. Initialise clique potentials, and then the network potentials by performing an initial propagation.

Besides forming a framework for exploiting local computations, the DAG can be a useful way of interpreting the (possibly complex) joint distribution (Jensen, 1993; Pearl, 1986a).

Such discussions typically centre upon what *effect* or *influence* observed variables will have on the remaining unobserved variables, or, in statistical terms, which variables are (perhaps conditionally) dependent or independent.

A variable, $A$ say, is said to *influence* another variable, $B$ say, if evidence applied at $A$ affects the belief in the outcome of $B$. How variables affect each other may depend upon which other variables are observed/unobserved. Three situations may be defined (Jensen, 1993), from which mechanisms of influence in the joint distribution may be (conceptually) understood:

1. Pipelined influence (figure 2.6(a)): two or more edges form a directed path. If only $A_1 = k$ is observed, then this may influence $A_2$ which in turn may influence $A_3$ and so on. Likewise, if only $A_n = j$ is observed, then this may influence $A_{n-1}$, $A_{n-2} \ldots A_1$. However, if $A_i = m$ is observed, then observing any one of $\{A_1, A_2 \ldots A_{i-1}\}$ will not influence any of $\{A_{i+1}, \ldots, A_n\}$. In this regard, $A_i = m$ is said to *block* the mechanisms of influence that originally existed between the variables $\{A_1, A_2 \ldots A_{i-1}\}$ and $\{A_{i+1}, \ldots, A_n\}$.

2. Diverging influence (figure 2.6(b)): all edges are directed away from a vertex. If vertex $B$ is unobserved, then vertex $A_j$ may influence vertex $A_k$ for all $j, k = 1 \ldots n$, $j \neq k$. If, however, $B$ is observed, all mechanisms of influence between the children of $B$ will be blocked by $B$.

3. Converging influence (figure 2.6(c)): all edges are directed toward a vertex. If $B$ (and any of its children if they exist) is unobserved, then the parents of $B$ cannot influence each other. If, however, $B$ (or any of its children) is observed, then the parents of $B$ may now influence each other (a point overlooked by Sucar and Gillies (1994)).

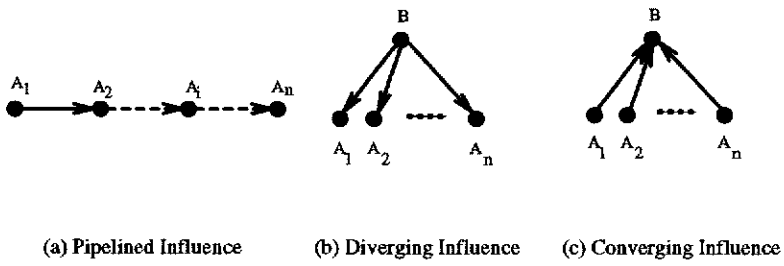



Figure 2.6: Mechanisms of Influence.

Blocking reflects the scenario that some pieces of evidence are more pertinent to a particular hypothesis than others. For instance, in figure 2.6(b), if $B$ is observed, then observing any of the children of $B$ tells nothing more about the other children.

Blocking in a general DAG can be considered to be made up of the above three basic components. Taking figure 2.5 as an example, if only $v_1$ is observed, then $v_1$ cannot influence $v_8$. That is, $v_1$ may influence $v_2$ (pipelined influence) which in turn may influence $v_4$ but not $v_3$ (because of converging influence). Vertex $v_4$ may influence $v_6$ but not $v_8$ (converging influence), so this path of influence is blocked. The only other path of possible influence is via $v_7$ (diverging influence), but as $v_3$ is not influenced by $v_1$, then this path is also blocked.

If, however, $v_6$ was observed, then $v_1$ may now influence $v_8$ (via $v_7$), as $v_2$ can now influence $v_3$ (converging influence — $v_6$ is a child of $v4$) and also $v_4$ may now influence $v_8$.

### 2.3.5 Bayesian Learning Strategies

Constructing a Bayesian network requires firstly that a model be chosen (network structure) and then that the conditional probabilities (parameters) be estimated in some way.

Assuming a network structure, one approach to specifying the rule-strengths (conditional probability tables) is to have experts give their intuitive feeling for what the rule strengths should be and then use these numbers directly. This

approach seems most appropriate when there are little data available for estimating the parameters directly.

A problem with the above approach is that it can easily lead to a network with probabilities that are intuitively in conflict with what would be expected (Osherson et al., 1994). Knowledge acquisition must be performed with care in such circumstances, and shows that modellers sometimes have difficulty deriving good probabilities from experts.

Where much sample data are available, using the data to estimate the parameters is appropriate. For example, if $A$ and $B$ are observed nodes, then estimating the parameters for $P(A|B)$ amounts to counting the number of times the configurations of the states are observed in the data and performing the calculations $P(A|B) = N(A,B)/N(B)$, where $N(X,Y)$ denotes the number of times the configurations of $(X,Y)$ appear in the sample data.

When the data are incomplete (that is the network has unobserved nodes) then an expectation maximisation (EM) algorithm may be appropiate (Dempster et al., 1977; Jordan and Jacobs, 1993). This is considered further in section 4.5.

The network stucture may be defined based upon subjective assessments of the relationships that are likely to exist, or automatically extracted (*learned*) using samples of domain (training) data (Badsberg, 1992; Hojsgaard and Thiesson, 1995; Heckerman, 1995b, 1995a; Heckerman et al., 1994), which is in general *NP-hard* (Chickering et al., 1994; Arnborg et al., 1987).

A commonly-used strategy in learning Bayesian networks is to start with a complete graph (every possible interaction modelled) and remove edges one by one until the new model created does not vary *significantly* from the complete model. Edges that add little to the model *statistically*, but significantly increase the computational cost of the model, are candidates for deletion.

## 2.4 Summary

An important outcome in the application of remote sensing and GIS technologies to broad-scale environmental monitoring is deriving accurate, cost effective information, typically presented as a classification. Achieving the desired classification accuracy may require that multiple sources of data be combined.

The general process of classification was considered in section 2.2, where perhaps the most popular approach, MLC, for the classification of optical remotely sensed data considered in isolation was described.

Noting that more than one source of data may be required, approaches adopted by other authors for combining data were discussed. Of the approaches, the use of expert system technology has appeal as it allows for the construction, maintenance, and relative ease of modifying potentially large sets of rules associated with combining multiple sources of data. Expert systems also have the appeal that they can be used interactively for decision support, answering what-if senarios.

For broad-scale monitoring, broad-scale data are required. Typically associated with such data is an element of data error, as the data are often collected with cost constraints, or collected with another purpose in mind (for example, the elevation data were never collected with the intention of mapping and monitoring salinity). Also, some data may not be available.

Based on the above points, expert system technology which allows uncertainty in rules to be specified is considered appropriate. Two competing theories (probability theory and the Dempster-Shafer Theory of Evidence) for representing and reasoning with uncertainty often appear in the Artificial Intelligence (AI) literature, along with much debate about the virtues of each. To better understand the debate, the two theories are reviewed in section 2.3, and some examples of their use for combining remotely sensed and GIS data are given.

The structure and calculations required for expert systems based on the competing theories are then reviewed. In particular, the approach of Shenoy and Shafer (1990) allows similar systems to be conceived using either approach to

uncertainty calulations.

Of the two approaches, the probabilistic method is to be adopted. This choice is made on the basis that specifying rule weights is typically a difficult problem and the weights are probably best estimated where possible; unlike the Dempster-Shafer Theory of evidence, probability theory has long provided guidelines for doing this. Another advantage of using probabilities is that if they are estimated from data, they represent relative frequencies which are particularly useful in the interpretation of remotely sensed and GIS data, where relative frequencies can be interpreted as proportions of a region having a particular class.

## 2.5 Deriving Landform Features

Landform is a significant factor in agricultural production, influencing the relative growth of crops within a single paddock as well as governing larger scale processes such as salinity and erosion. The potential for using digital elevation models (DEMs) in an agricultural geographic information system (GIS) appears great.

The focus of this section is to review algorithms that can be used to process the DEM in such a way that meaningful results for agricultural GISs are extracted. In section 2.5.1, issues relating to elevation data are discussed. In section 2.5.2, some potential applications are introduced and then in section 2.5.3, algorithms for processing DEMs are considered.

### 2.5.1 Background

The major source of elevation data in W.A. is digitised contour levels obtained by stereo photo interpretation. Depending upon the study area, contour level increments no better than 5m (for example in the Kent catchment), 10m (for example the region around Dumbleyung) or 20m (for example the eastern extent of the MKG study area) may be the best available.

Digital elevation models (DEMs) can be constructed from the data using a

GIS package such as GRASS 4.1. A DEM may be represented as a raster map, where each pixel has a value equal to the elevation at that cell. To form a DEM from contour data requires elevations at points of missing data to be interpolated from known elevation data points.

Typically, much of the West Australian wheat belt has a gradient of less than four degrees ($4^o$). This, coupled with a 10m elevation (contour) sampling interval, means that contour lines are commonly spatially separated by distances greater than 140m. For areas with slope less than $1°$, this equates to spatial separation not less than 500m.

A consequence of the 10m elevation sampling interval is that it is possible for the landscape to undulate within $\pm$10m of the current elevation level and only be sampled a few times. Assuming a slope less than $1^o$ say, then this effect can mean that there are no elevation samples over large areas for which significant landform changes have occurred. This effect is often observed in practice.

Consequences of the above are that changes in landscape of local significance may be completely unobserved in the DEM data. For instance, breaks in slope are not observed, as are features that have elevation changes that fall within a contour level increment.

For example, consider a broad valley system bounded by land of slope less than $1°$. Further assume that the valley floor is flat and its boundary is defined where the slope of the land changes from $0^o$ to $1^o$ (this is not an unrealistic approximation to the true event). Then, assuming that the valley floor elevation is just above the last contour sampling height, the next elevation sample will fall approximately 500m from where the break of slope is actually located. Interpolating this contour will produce a relatively flat area within its extents. Thus a relatively flat area bounded by $1^o$ slopes is generated, but the break of slope (as would be identified by an automated technique) will be displaced by hundreds of metres from its true position. This scenario is depicted in figure 2.7.

These simple considerations of the source of the elevation information give

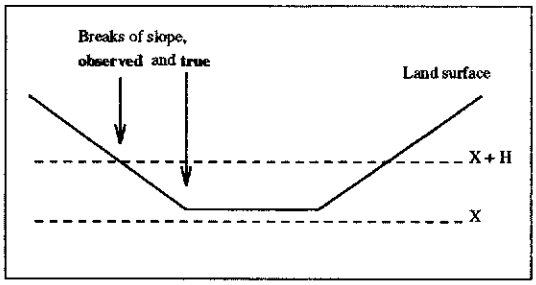



Figure 2.7: Effect of Quantisation Errors.

valuable insight into the theoretical limitations of any DEM derived from contour data alone.

Another aspect of the DEM generation process that affects feature extraction algorithms is how to deal with local minima that may be a consequence of the interpolation process and not a true depression. Such depressions are commonly refered to as *pits* and have great influence on drainage network extraction algorithms. A common approach for handling such problems is to *fill* the pit (that is, modify the DEM) (Vincent and Soille, 1991) in much the same way as a hole in the ground would be filled: increase the elevation of the area of the depression until no depression exists in the modified DEM. Thus a pit becomes a flat area.

Flat areas pose problems for algorithms that use parameters such as slope and curvature calculated from local neighbourhood operations. Apart from the case that an area is flat, flat areas in a DEM may be the result of pit filling as described above or result from poor interpolation in large areas possessing little elevation data.

Preprocessing techniques that modify the DEM so that flat areas (in the strict sense of equal elevation values) do not exist can be employed to avoid such complications. A strategy commonly employed is (Vincent and Soille, 1991):

> *For each flat region, perform a distance computation so that each pixel in the flat region has a value equal to its distance from the closest boundary pixel that possesses a neighbouring pixel of lower elevation. Then, for each pixel in the flat region, slightly modify its elevation value*

*by adding the scaled computed distance value to its elevation.*

For example, all pixels neighbouring a flat hilltop have elevations lower than the hilltop, so the above operation calculates for each cell in the flat region the distance the cell is from the region boundary, with the centre of the region having the greatest distance. When these distances are added to each cell's elevation, a hilltop is mimicked with the centre of the region being the hilltop maximum.

When performing the same operation on flat valley segments, a ramp is generated whereby the ramp slopes upward from where the valley segment would drain.

The use of cubic spline interpolation and scaled elevation values (elevation expressed in centimetres) largely avoids flat areas of exact equal height being generated, so that the previous DEM modification techniques often do not need to be employed.

### 2.5.2 Potential Applications for Digital Elevation Models in Agriculture

One of the simplest applications of DEMs is to use the raw elevation data along with simply derived data such as slope, aspect and curvature estimates as predictive variables for events such as salinity, waterlogging and wind erosion.

Another approach is to consider the behaviour of landform features. When discussing agriculturally related issues, terms such as ridges, valleys, hilltops and breakaways are commonly used (Ferdowsian, 1993). These features relate expert understanding of certain processes to what is observed, and provide a natural partitioning of the landscape. It would therefore seem beneficial to be able to extract landform features from the DEM for the purpose of providing support for higher level reasoning processes. In this role, the features provide a description of the landscape that can be operated on within a GIS environment. Features that have been automatically extracted by other researchers include: broad, flat

areas of terrain (Chorowicz et al., 1989); hills and plains (Graff and Usery, 1993); peaks, pits, ridges and ravines (Kweon and Kanade, 1994); ridges and valleys (Riazanoff et al., 1988); sharp valleys, wide valleys, summit, plateau (Chorowicz et al., 1992); and drainage networks (Qian et al., 1990; Chorowicz et al., 1992; Jenson and Domingue, 1988).

Apart from providing features for classical GIS type manipulations, DEMs also offer a means for simulating hydrological processes that occur across the landscape; for example, models that trace the flow of water across a terrain surface after a simulated rainfall event (Quinn et al., 1991). These models are typically called *water accumulation* models. In their most simplistic form, these models make two assumptions: (1) water flows down hill; and (2) all water flows on, or close to, the DEM surface.

Information such as landcover can be incorporated into the models so that factors such as plant water interception (uptake) can be included in the model (Jenson and Domingue, 1988; Schultz, 1994). The volume of water flowing across the landscape, landcover and gradient of the landform may also give indications of areas of potential water erosion.

Karnieli et al. (1994) present a model for catchment scale runoff prediction. The model considers the catchment as being partitioned into a number of (connected) sub-catchments that have parameters relating to surface runoff. Their paper considers the calibration and testing of a forcasting model for performance on two objectives: runoff volume and peak discharge. Such models rely on catchments and sub-catchments being defined. Algorithms to extract (sub)catchments typically come under the name of *watershed* algorithms.

Algorithms relating to the diverse use of DEMs are discussed in the next section.

### 2.5.3 Processing Techniques

This section discusses processing techniques for DEMs relevant to some of the application areas discussed in the previous section. The algorithms are discussed in the following order: section 2.5.4 covers landform feature extraction, including valley segments, ridge segments and hilltops; section 2.5.5 considers water-accumulation algorithms, and section 2.5.6 covers drainage network and watershed extraction algorithms.

As will be apparent, many of the above sections have overlapping concepts and hence processing techniques.

### 2.5.4 Landform Feature Extraction

#### Statistical Approaches

Many early methods of extracting features from DEMs concentrated on deriving variables relating to surface properties such as slope and curvature (first and second derivative of the surface respectively) and then attempted to classify each pixel based on these parameters. For example, if $z$, $z'$ and $z''$ represent elevation, slope and curvature respectively, then a peak would be defined by $z' = 0$ and $z'' < 0$. Haralick used these principles to detect ridges and valleys (Haralick, 1983) and drainage networks (Haralick et al., 1985).

With this approach, the labelling of a pixel ignores the results of the labelling of neighbouring pixels, leading to isolated pixels being classified as a ridge (valley) and disconnected ridge (valley) lines. Estimating $z'$ and $z''$ across larger areas may also pose problems, as a choice of appropriate neighbourhood size must be made.

Chorowicz et al. (1989) present an algorithm for extracting geomorphological features such as crests, thalwegs and various slope changes. These sets of primitives were then used to identify boundaries relating to strike ridge morphology, fluvial deposits in valley bottoms, crest lines and drainage patterns.

The algorithm is based on a *search for strings* approach. In this approach, the

DEM is first processed to form four new data layers based on independently scanning the image in the N-S, E-W, NW-SE, and NE-SW directions. The scanning process produces *profile* images composed of the symbols $+$, $-$ and $=$ for each scan direction. If a buffer $F$ holds the elevation data for the current scan direction of scan length $n$, and the buffer $S$ the results of the scan, then the symbols are assigned as follows:

- $S(i) = \ '+'$ if $F(i+1) > F(i)$.
- $S(i) = \ '-'$ if $F(i+1) < F(i)$.
- $S(i) = \ '='$ if $F(i+1) = F(i)$.

where $i = 0, 1, \ldots, n-1$.

Intuitively, $S = [+, -, =]$ means the profile is uphill, downhill and flat respectively in the direction of the scan.

Each scan line of each of the profile images is searched for patterns that define the profiles of the features being searched for. For example, if along a scan line $S$ the profile changes from $+$ (uphill) to $-$ (downhill), then this is evidence of a local maximum in this scan direction. Some of the profiles defined by the authors are:

- $A = [-- == ++]$ (flat-bottomed valley).
- $V = [- - ++]$ (incised valley).
- $E = [++ == --]$ (plateau).
- $S = [+ + --]$ (crest or summit).
- $E1 = [++ == ++]$ (upgrade horizontal step).

In this way, four images composed of feature profile symbols including those mentioned above are independently constructed for each scan direction. Then, at each position in the landscape, a decision is made as to the true feature label based

on the four feature profile symbols. For example, if at pixel location $(x, y)$ all the profile images had symbol $S$, then the symbol $S$ would be a natural choice for the final label. The choice of profile symbol becomes more interesting when obscure mixtures of these symbols occur. The authors use a second derivative test in the combination of symbols phase, but perhaps the key element of the approach is the profile scanning described above, as ultimately it limits the accuracy of any further calculations. Some comments on this method are:

- Evidence of features of greatly different spatial extents can be easily collected.
- The algorithm uses simple operations and hence is computationally efficient (O($n$) where $n$ is the number of image pixels).
- Some spatial strength is present in the determination of variables. For example, using this method, the profile of a flat bottom could be defined as a variable length string of '=' symbols bounded by the strings '−−' and '++'. If such a string of '=' is detected, then all symbols in this string constitute a flat bottom.

In practice, this technique performs poorly because noise and even relative orientation of the features in the DEM affects the labels in the +,−,= images resulting in bogus labels in the independent symbol images. This causes the final labels to be inconsistent with their neighbours. Combination of the many different label combinations also poses problems. This point highlights the spatial weaknesses of the approach.

In (Chorowicz et al., 1992), this technique was incorporated into a drainage network extraction algorithm. The primary role of the *scan-for-strings* approach was to identify flat basins, an important feature in the Western Australian wheat belt. The application of this technique to wheatbelt data is considered further in section 5.4.1.

### The Structionalist Approach

Features such as ridge lines are essentially connected components and extracting them as such is desirable. A common technique to ensure connectivity is to build on what is presently known using a local searching mechanism. For example, if $f(x, y)$ is considered part of a feature being extracted, then the neighbours of $f(x, y)$ are examined to see if they also have the property of belonging to the feature. If so, the neighbour is added to the feature and its neighbours are checked and so on. This sort of mechanism requires a starting point (or feature segment) as well as a local search constraint that defines the feature.

Riazanoff et al. (1988) refer to the above approach as a *structionalist* approach. They examined this approach for extracting ridge and valley lines. Critical points of the DEM surface including local maxima, local minima and saddle points were used as the search starting points. Search constraints such as *descend in the direction of steepest slope* were used as local search criteria. For example, an algorithm that used saddle points and the search criteria *ascend in the direction of steepest slope until reaching a local maxima or image edge* was used for the purpose of extracting ridge segments. Larger scale ridge lines are possibly made up of a number of these segments, although grouping of segments was not discussed in their paper.

Using the structionalist approach, neighbours of saddle points are commonly used as candidates for *channel heads*. Given the channel heads, the drainage network is constructed using the search constraint *descend along the path of steepest slope*. The descent stops when either a previously defined channel, a local minimum or the edge of the image is found. Examples of this approach can be found in (Riazanoff et al., 1988) and (Chorowicz et al., 1992).

The key features of the above approach are:

- The algorithms rely on accurate identification of surface critical points. This may not always be possible as, typically, simple neighbourhood masks are used for this purpose.

- Where variables governing the local search constraint are poorly defined, so is the resulting extracted feature. This is particularly apparent when trying to use slope criteria in relatively flat regions.

At this point, it is worth making a distinction between the detection of drainage / valley lines and the extraction of drainage networks. The concept of a drainage network is far stronger, as it is a collection of connected drainage / valley segments such that (surface) water flow constraints are enforced. For this reason, drainage networks will be discussed separately in section 2.5.6. Before leaving this section, algorithms that are perhaps more structured in their approach to feature extraction are examined.

### Morphological Approaches

Kweon and Kanade (1994) take a contour based approach to extract peaks, pits, ridges and ravines (valleys) from elevation data. The inspiration for their approach is that human experts can interpret these types of feature from contour data alone. They also point out that their method does not suffer from the locality problems associated with some of the previously mentioned techniques.

The authors define peaks and pits as being composed of a series of closed contours, with the direction of ascent distinguishing the two. Valleys and ridges are described as contour lines that form smooth and rounded $V$ shapes (see figure 2.8), again with relative contour heights distinguishing the two.

The algorithm firstly constructs a contour map from the gridded elevation data, where contours are extracted at regular intervals, $H$ say, in height. Contours are produced in polygon form.

From the contour map, a contour tree is constructed. A contour tree represents the topological relations that exist between the contour lines of a map. The tree (graph) has a vertex for every closed contour and an edge between vertices that has the property that the two contour levels are only $H$ different in height and their corresponding closed contours have an inclusion relation, that is one contour

region falls completely within the other. In this way the tree branches whenever the landscape splits into two or more hills. Leaf nodes of the tree represent contour levels defining a local maximum.

The contour tree is constructed by the following method:

- Create a node for the lowest contour level. This node becomes the root of the tree, and w.l.o.g assume that the corresponding contour forms one connected component.

- Proceed to the next height increment and extract all connected components. Compare these components with those from the previous height for the following conditions:

  Two or more components at the new level are included within a single component from the previous level. In this case the terrain has *split* into two or more new hills. A new vertex and edge is added for each new *hill.*

  There is no component in the new level that falls within a contour of the previous elevation. In this case, a local maximum has been found.

  There is only one contour at the new level that falls within a closed contour of the previous level. In this case, we are still proceeding up the same hill, and a vertex and edge are added for this situation.

The tree is constructed via a depth-first search strategy. An example of such a tree is given in figure 2.9.

Peaks can then be extracted using a graph traversal technique. The technique starts at the leaf nodes (c1 and d in figure 2.9) and searches back towards the root node, stopping when a topologic change is encountered. The path traversed is then given a unique label. In the above example, c1 defines a peak as does {d,c2}. Pits can be similarly extracted by flipping (inverting) the data about the horizontal axis and performing the above operations.

The principle underlying the extraction of ridges and ravines in this method is to find local maxima of the curvatures of the contour lines and then group the

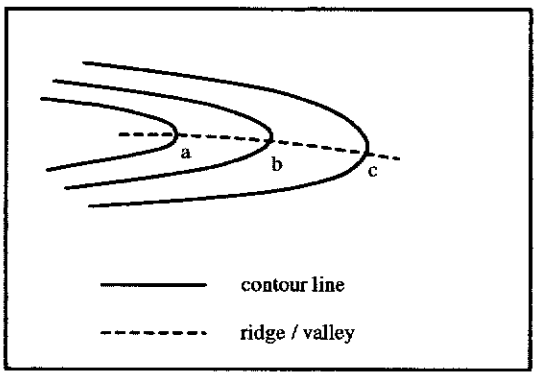



Figure 2.8: Ridge and Valley Descriptions.

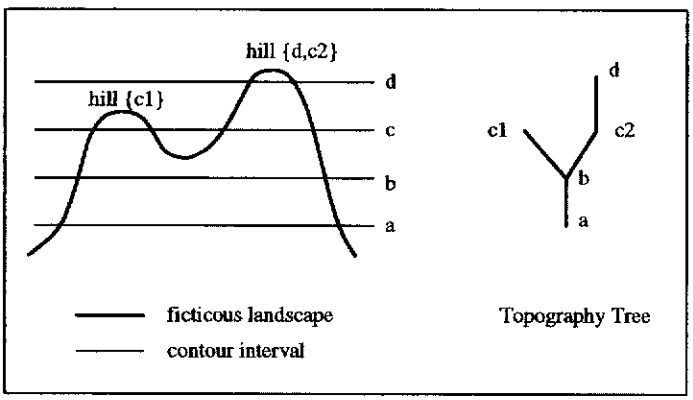



Figure 2.9: Terrain and Associated Topography Tree.

maxima together to form ridge and valley segments. This simple idea is depicted in figure 2.9, where points $a$, $b$ and $c$ represent local extrema in $z''_t/z'_n$ of the contour line.

In practice, the local extrema are identified by using an arming method at multiple resolutions, where resolution proceeds from coarse to fine with the aim being to successively narrow down the position of the extrema. The arming method is shown pictorially in figure 2.10 and the curvature $c_i$ at $P_i$ is calculated from

$$c_i = acos(v_{i-k,i}.v_{i,i+k}/|v_{i-k,i}|.|v_{i,i+k}|)$$

where $v_{i,j}$ is a vector from point $P_i$ to $P_j$.

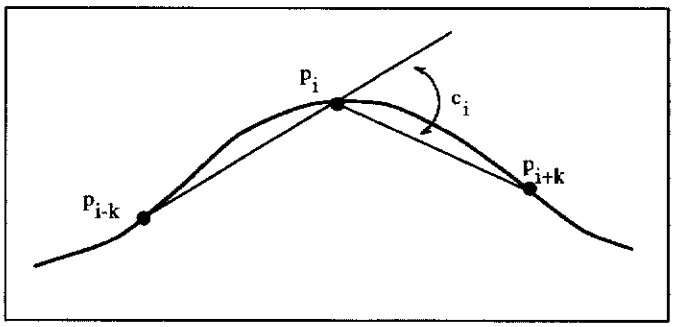



Figure 2.10: Arming Method.

The arm length $k$ controls the amount of smoothing in the calculation, with smoothing increasing with $k$.

Once the extrema have been identified, the next step in the algorithm requires that these points be linked to form ridge and ravine lines. This step makes use of the topologic tree formed earlier. The nodes of the tree define the extrema points that are topological candidates for linking.

The linking proceeds as follows: traverse the contour tree in breadth first fashion. For each candidate ridge (ravine) point at the current node, $N$ say, all ridge points defined at the child nodes of $N$ become candidates for linking. As many such children may exist, the *best* candidate is chosen. The criteria for *best* is derived from measuring variables such as the local direction of the new node as

opposed to the global direction of the ridge (ravine) already formed.

Attributes such as global current ridge direction, local current ridge direction, euclidean distance and curvature similarity are considered when linking points. For example, only points within $\pm 90^{\circ}$ of the local and global directions are considered as candidates for linking. The above attributes are combined to form a heuristic *similarity* function that is used to decide on the *best* point for linking. On an intuitive level, this function constrains the forming ridge line to proceed in a *reasonable* direction based on the segment of ridge that has currently been formed. Valleys are similary contructed.

As is evident from the above discussion, the choice of $H$ greatly affects the size and shape of the pits and peaks that are extracted, and must be chosen with some care.

Peaks and pits are defined by a series of closed contours. Strong non-local topological constraints are used to define and extract these features. These definitions and constraints are clearly superior to other peak-classifying techniques such as those presented by Graff and Usery (1993) which use a gradient change to define peak boundaries and then local operations starting from critical points such as local maxima to flood-fill the resulting shapes. The weaknesses of the Graff and Usery (1993) approach was evident in the authors' results showing that only 16% of peaks were well defined for a sample of relatively flat undulating terrain (which represents most of the West Australian wheat belt).

Using the model proposed by Kweon and Kanade (1994), there is no guarantee that ridge (valley) lines will be situated precisely on the local maxima (minima) when overlaid onto the original DEM, as they are interpolated from local contour extrema. From an application perspective, ridge and valley lines extracted by this algorithm merely serve as indicators of local terrain maxima and minima, and have no stronger connectivity properties for defining features such as watersheds and drainage networks.

Another method of expressing a terrain's local maxima and minima properties

is to simulate water flowing across the terrain surface and then stratify the results into various water accumulation levels (Jenson and Domingue, 1988). This strategy is a by-product of performing a water-accumulation calculation and observing that hill-tops and valleys are represented by low and high levels of water accumulation respectively. This class of algorithms is introduced in the next section.

### 2.5.5 Water Accumulation Algorithms

After processing a DEM with a water-accumulation algorithm, each cell in the resulting image has a value that represents how many units of water have drained to that cell. The simple assumption that water flows down-hill across the surface of the DEM is generally used, although this assumption may in some cases be relaxed. For example Quinn et al. (1991) considers inferring from the DEM the subsurface hydrological properties of an area and then applying the water accumulation algorithms to the inferred hydrological surface.

Let $A$ denote a water accumulation image. The algorithm starts by assigning to each element of $A$ the amount of "rain" that fell on that cell. Quite often, all cells are initialised to have one unit of rainfall.

The algorithm then proceeds to pass water down-hill until the landscape has drained. In practice this is carried out by each element (pixel) of $A$ passing water to its downhill neighbouring pixels, and these pixels passing the water downhill to their neighbouring pixels and so on. Each cell maintains a count of how many units of water drained to it. A landscape has drained when there is no more water to pass down-hill. Pits do not drain as by definition no downhill neighbours exist, hence all water flowing into a pit is trapped. This has big implications for cells further down the landscape, as they never receive the water that was trapped by a possibly false local minimum. Pit-filling techniques are often used to modify the original DEM so that flow is maintained.

Perhaps the biggest difference that exists between published algorithms (Jenson and Domingue, 1988; Quinn et al., 1991) is the choice of how water passes

downhill. In this context, algorithms may be classed into *single-outflow* and *multiple-outflow* algorithms.

Single-outflow algorithms consider only the downhill neighbour possessing the greatest drop in elevation as the route that water will flow. That is, all water at a pixel is passed to a single downhill neighbouring pixel (if one exists).

Multiple-outflow algorithms distribute the water amongst all downhill neighbouring pixels. Typically, each downhill pixel receives a portion of water proportional to the drop in elevation to that pixel.

For relatively flat terrain where drainage direction is poorly defined, multiple-outflow algorithms appear advantageous (Quinn et al., 1991).

In applying these algorithms to real data, a number of problems are often encountered. The first problem is what to do with spurious pits. In the following it is assumed that pits have been filled with an algorithm such as described by Vincent and Soille (1991). The second problem of major importance is how to treat large flat areas where drainage is either poorly defined or not defined at all. Examples of large flat areas include plateaus (flat hill-tops and ridges) and broad flat valley systems, with the latter example being a common feature of the W.A. wheat belt.

In this discussion, a large flat area will be defined as a connected component of cells whereby each cell has elevation equal to or significantly close (such that it can be considered equal) to that of its eight neighbouring pixels.

By definition, water cannot possibly flow down-hill when considering a local neighbourhood within these regions. These regions may be present within the original elevation data or the result from a filled pit.

One technique is to modify the elevation values of all cells within these regions as described in section 2.5.1.

Another technique relies on assigning flow directions for each cell in the DEM, the approach taken by Jenson and Domingue (1988). This technique does not rely on flat areas being explicitly defined. The algorithm proceeds as follows:

1. Pixels are defined to flow to the neighbouring pixel having the greatest drop in elevation, if such a neighbour exists. If such a neighbour does not exist, that is all neighbours have greater than or equal elevation, then no flow direction is defined.

2. For each pixel with no defined flow direction, label it as flowing to a neighbouring pixel with a defined flow direction (if one exists). Only neighbours that do not flow to the pixel in question are considered. This operation is iteratively applied until all pixels have a defined flow direction.

The net effect of this is that direction assignments iteratively grow into the flat area from the outflow points until all pixels have flow directions assigned. The single-outflow water accumulation model can then be applied using these flow directions.

The above-mentioned technique constructs models that allow water to flow unimpeded across the (perhaps modified) DEM surface. Where surface flows are not defined, the models work to define a flow direction for each pixel. This raises an important issue: is it justifiable to define for each pixel in a broad-valley system a flow direction when the data essentially provide little or no information as to what the flow for each individual pixel should be?

Given that the flow direction for each pixel was inferred by considering the flow characteristics of the larger connected region, would it not be better to consider the larger region as an entity in its own right and work with the (defined) characteristics of this region rather than trying to extract some form of artificial precision by considering each element of the region separately. This point is considered further in section 5.5.

### Water Accumulation Model Extensions

If the simulated rainfall event initialises each cell of landform to receive one unit of rain, then each cell in the resulting water accumulation map represents the area

of land *up-slope* from that cell. This variable can be used as a terrain descriptor.

The accumulation models discussed above simulate water flowing across the terrain surface. Obviously, many factors including land cover, soil type and contour banks will affect this flow. Schultz (1994) discusses the role of remote sensing in hydrogeological monitoring and suggests that remote sensing input can be used both as a basis for model parameter estimation and also to provide model input.

Examples of using Landsat TM data include calculating parameters relevant for evapo-transpiration, water interception and infiltration rates estimated on the basis of landcover as well as a vegetation index, such as the *Normalised Difference Vegitation Index* (NDVI).

Also, other hydrogeologically relevant information such as land slope (relevant for flow velocity) or exposure to the sun (based on aspect and relevant for evapotranspiration) can be extracted from the DEM. These parameters are then incorporated into the accumulation models to refine the flow characteristics.

Other variables such as water erosion risk, which is a function of variables including volume of flow (water accumulation), flow velocity and landcover, can also be calculated. Packages such as ARC/INFO and GRASS have incorporated such algorithms, with the single outflow models being the basis of the calculations.

Schultz (1994) also makes the important point that hydrogeological processes on different scales require different models and different scales of analysis. For example, monthly rainfall volumes were used to predict monthly river runoff for the purpose of water resource planning. Such models require water-sheds and drainage networks to be defined. This topic is the subject of the next section.

### 2.5.6 Drainage Networks

According to Chorowicz et al. (1989), a drainage network is a system which is favourable to hydrogeological flow and is based on the following fundamental properties:

- Ideally, most drainage channels start from the close neighbours of saddle points.
- At each point of a channel, hydrogeological flow follows the direction of steepest slope.
- Drainage channels do not cross one another. At a meeting point of two or more channels, they join and continue as one channel.
- Hydrogeological flow continues until it gets into a depression without surface outlet, or otherwise disappears by mechanisms such as infiltration and evaporation, or until the flow leaves the region being observed.

Based on the above definitions, an algorithm firstly identifies channel heads close to saddle points and then, using these points, descends in the direction of steepest descent to extract skeletal drainage networks. Then broad valley features are extracted by a flat area classifier and added into the networks by simple masking.

The water accumulation algorithms described in section 2.5.5 can be used directly to extract drainage networks by thresholding the accumulation image (Jenson and Domingue, 1988). The lower the threshold, the denser the network. The results from the single-flow procedure will produce skeletal tree networks whereas the result from the multi-flow procedure produces a *fuzzy* network.

A totally different method for drainage network extraction is presented by Qian et al. (1990). The algorithm is based upon using high-level reasoning to link low-level derived attributes into a *consistent* drainage network. Ridge line extraction was also considered.

The high-level reasoning mechanism uses the Dempster-Shafer (DS) theory of evidence to guide the linking process, based on evidence produced from *evidence* tests. Uncertainty is incorporated into the reasoning process.

The algorithm starts by using a local neighbourhood to look for $V$ and $A$ type shapes (similar to those described by Chorowicz et al. (1989)) that are

indicators of a valley segment, with the result being a set of partially connected valley segments. It is then the job of the expert system to link (or split) segments so that a consistent drainage network is derived.

The expert system combines evidence generated by a series of 8 tests that include the relative proximity of segments, curvature similarity of segments, segment length test, relative slope tests, relative elevation tests and ridge intersection tests. Nine thresholds are used to map test measurement variables to a corresponding number of Yes/No answers. Probability distributions are then defined to map these Yes/No answers to a hypothesis set. That is, if $H$ represents the set of hypotheses and $T = \{t_1, t_2, \ldots, t_8\}$ represents the set of tests, then the evidence strengths are defined as $p(H|t_1)$, $p(H|t_2), \ldots,$ $p(H|t_8)$. These values are determined from conditional probabilities or subjectively by experts.

An important point to note with the model is that valley segment joining indicators are being combined with valley constraints using uncertain reasoning. A potential problem with this approach is that a constraint is treated as a variable that exhibits uncertainty. Since DS, as implemented by the authors, essentially combines all variables as though they are independent, then constraints may not be satisfied in the final decision, as the evidence from the constraint may get *weighted-out* by the evidence from the other (indicator) variables. Constraint variables in this case would need to be specified with certainty 1.

Another point to note with the algorithm is that when it is decided to connect valley segments, the segments are connected by an interpolated cubic spline. There is no local control used in the placement of the spline so it may not lie in the valley bottom.

The authors also report that errors at the high-level reasoning stage are made, resulting in valley segments crossing ridge lines. A verification stage is given to resolve this problem, which essentially looks for ridge line pixels intersecting valley pixels. Note that it appears that test 8 (which is a constraint to avoid this condition) has failed, possibly because the constraint was *weighted-out.*

From the results presented, the simple steepest-descent algorithm appears to be superior to this method.

The authors also suggest that drainage basins could be determined using the drainage channels and ridge lines of this method, although given the above comments, this would not seem to be practical. The extraction of watersheds and drainage basins is discussed in the next section.

### Watersheds

Many published algorithms used to extract watersheds rely on local search techniques, such as those presented in section 2.5.4, to identify dividing lines such as ridge lines and then attempt to connect these features into geomorphological networks. As mentioned in section 2.5.4, problems exist with these approaches when the local terrain is poorly defined, for example in areas that are relatively flat. As the landscape considered in this thesis is relatively flat, undulating terrain, these algorithms will not be considered as candidates for water-shed derivation.

Two differing techniques that show promise are those based on flow directions and those based on immersion simulations.

### Flow Direction Technique

It is assumed here that a flow direction image has been created from the DEM using the process described in section 2.5.5 and that an accumulation image based on this flow direction image has also been created. To recap, each pixel in the flow direction image has been assigned a direction(s) in which water will run from this pixel. The accumulation image expresses how many pixels (where we are assuming initially that each pixel had one unit of water) would drain to a cell after a simulated rainfall event. In this context the accumulation value also represents upslope area. In creating these images, it is important that the role of real local minima (such as lakes) be defined.

Extraction of watersheds requires that the flow direction data and another data

set that specifies the *base* (where all water in the watershed eventually flows to) of the (sub)watersheds be specified. Starting at a base, a (sub)watershed is created by iteratively labelling pixels as belonging to the base to which they (eventually) flow. That is, label the base with a unique watershed identifier, then label (recursively) all neighbouring cells that flow into the currently defined watershed as belonging to the watershed.

Automated techniques exist for extracting bases. For example, the terrain can be partitioned into sub-watersheds, where each sub-watershed must have an area greater than a specified threshold, by the following algorithm:

- Define a sub-watershed minimum area threshold, $T$ say.
- Using the accumulation image, calculate for each pixel a value $\delta$ which subtracts the accumulation value for a pixel from the accumulation value of the pixel ($c$ say) it flows to (as defined by the flow direction image). If the value of $\delta$ is greater than $T$, this indicates that there is other significant inflow into $c$, and hence $c$ represents the point where two tributaries converge.
- A pixel becomes a (sub)watershed base if both its accumulation value and its delta value are greater than the sub-watershed minimum area threshold.

The properties of watersheds extracted by the above algorithm are governed by the properties of the flow direction map, which may be poorly defined in flat areas of landscape (see section 2.5.5).

### Immersion Simulation Techniques

Immersion simulation techniques such as that presented by Vincent and Soille (1991) construct watersheds by extracting connected components of equi-height cells and then use inclusion properties of connected components at adjacent elevations. In this way, the use of possibly unreliable pixel neighbourhood properties (such as flow directions) to define watersheds is avoided.

The technique considers a DEM as a three-dimensional surface that has holes punched through all its local minima. These minima represent what the algorithm considers to be the watershed bases. In a conceptual sense, watersheds are constructed by lowering the surface into a fictitious body of water and observing from which minima (base) each part of the DEM surface is flooded from. A dam is built where water emanating from different minima meet. On completion, these dams define the watershed boundaries.

From an algorithmic viewpoint, the basis for identifying watershed boundaries lies in examining the relationships between connected components of pixels at height $i$, say, and height $i+1$ in the DEM. If $Y = \{y_1, \ldots, y_n\}$ and $X = \{x_1, \ldots, x_m\}$ represent the set of connected components at level $i+1$ and $i$ respectively, then three possible relationships exist between $Y$ and $X$:

- An element $y_i$ does not intersect any element of $X$. In this case $y_i$ is a new minimum.
- An element $y_i$ intersects only one element of $X$, $x_j$ say. In this case $y_i$ is considered to belong to the watershed represented by $x_j$ and is labelled as such.
- An element $y_i$ intersects more than one element of $X$. In this case, dams must be constructed to separate the elements of $X \cap y_i$.

The purpose of the dams is to partition $y_i$ such that each partition of $y_i$ can be considered as originating from a unique component of $X$ and hence as belonging to a unique watershed.

When a dam is to be placed on a flat area of terrain, for example a flat hilltop, distance considerations are used to position the dam. Basically, the net effect of the distance considerations is that dams are positioned in the "middle" of the dividing flat area.

The most important issue that needs to be addressed when using the above algorithm is what constitutes a watershed base, and how are these bases extracted.

Unless the DEM has been adequately preprocessed to remove all spurious local minima, then using extracted local minima will result in an over-segmented image.

### 2.5.7 Summary

Landform has a significant influence in the management of land, affecting smaller-scale issues such as the relative growth of crops within a paddock and broader-scale issues such as the management of surface water runoff which extends beyond a single paddock/farm.

The approach taken here is not to attempt the difficult (intractable) problem of providing a complete physical description of the processes occurring within the landscape, but rather to look at approaches that characterise the land in terms of different parameters relating to landform morphology. This approach is adopted since broad-scale information on the elevation of the land is readily available.

The data considered for the DEM calculations are derived from photo interpretation and consist of elevation contours sampled at regular elevation intervals (typically 5, 10 or 20m intervals in elevation). Given these data, limitations such as the inability to accurately identify the position of breaks (significant changes) in slope are discussed.

With these limitations in mind, published techniques that use the morphology of the land to make landform decriptions have been reviewed. Such approaches typically use descriptions such as *hilltop* and *valley* to describe the land.

Early published techniques were largely based on slope and curvature estimates on a per-pixel basis and then using approaches such as the MLC to classify the data into various landform classes. Obtaining good curvature estimates and spurious mislabelling of pixels, resulting from the per-pixel nature of this approach, are the main limitations.

Based on this, researchers moved from the per-pixel approach and used algorithms based on performing a local search for particular features. The problem of identifying initial search starting points remains.

Stronger morphological approaches have evolved, although many are based on providing the user with a conceptual description of the terrain rather than with variables that describe each location within the terrain. For example, algorithms for extracting skeletal descriptions of drainage systems received much attention in the literature.

Of the algorithms considered, those based upon simplistic assumptions about the land hydrology (for example, that *water runs down hill*) appear the most applicable for the mapping and monitoring problems considered in this thesis, as the *true* hydrology is their direct cause. These algorithms are typically called *water accumulation algorithms* and this methodology is adopted in this thesis.

## 2.6 Summary

This chapter reviewed literature on two important topics for the applications considered in the remainder of this thesis; firstly, the need to combine multiple sources of data to achieve adequate (or better) interpretation accuracies, and secondly the derivation of variables from digital elevation models for combining with other sources of data.

Within this domain, previous mapping efforts were largely based on using remotely sensed Landsat TM data, and the problem of representing and estimating the spectral characteristics of the different landcover classes has been considered at length by other researchers. The classification technique typically employed by these researchers is MLC, based on modelling the spectral characteristics of each class with a multi-variate Gaussian density as outlined in section 2.2.2. Based on the relative success in earlier uses of this approach, it was of interest to examine different ways for combining information obtained from this approach with other sources of information.

There exist a multitude of ways of combining remotely sensed data with other sources of data. Section 2.2.3 provides a conceptual overview of some of the dif-

ferent approaches. Expert system technology has been adopted as it provides a means of specifying, maintaining and modifying potentially large rule sets. Typically, not all data sources are available for a particular area, and of the data available significant errors are expected. Because of this, expert system technology which accommodates uncertain reasoning has been chosen. A number of alternatives for performing uncertain reasoning exist, with the two most common approaches in the remote sensing and GIS domain being probabilistic approaches and approaches based on the Dempster-Shafer theory of evidence. A probabilistic approach is the preferred option; in particular, Bayesian networks have been chosen, as they provide the basis for recent developments in probabilistic expert systems. With such systems, rules are specified as a set of conditional probability statements, for which, unlike methods based on the Dempster-Shafer theory of evidence, methods for estimating rule weights are well established in the statistical literature.

The Bayesian network approach also has the appeal that it is theoretically consistent with the MLC, which is itself a Bayesian classifier.

The second part of the chapter reviews methods for extracting features from digital terrain models. Early methods relied on obtaining slope and curvature estimates either to directly perform a classification using the MLC or to identify critical surface points for initialising local searching strategies. Noting difficulties in obtaining good slope and curvature estimates, researchers moved onto approaches which had stronger spatial emphasis and typically worked directly with elevation. Of the approaches, those that simulate hydrogeological processes appear advantageous as the hydrology has a direct influence on many of the mapping and monitoring problems considered here.

# Chapter 3

# Problem Domain

## 3.1 Introduction

As defined by Ferdowsian (1994),

> *A soil is considered to be saline if it contains sufficient soluble salts to limit plant growth and will become a problem when production is affected and/or soil structure and stability are endangered.*

In much of the Western Australian wheatbelt, the clearing of land for agricultural production has led to rising saline groundwater, resulting in the loss of previously productive land to salinity.

This form of salinity is referred to as *dryland salinity* and is distinct from other forms of salinisation due to factors such as irrigation. Only dryland salinity is considered in this thesis.

Based on farmer surveys conducted in 1979 and 1989, the Australian Bureau of Statistics reported that 443 000ha (2.8%) of previously arable land was lost to salinity, at a rate of about 18 000ha each year. Salinity which was present before, and which has occurred after, European settlement is termed *primary salinity* and *secondary salinity* respectively.

The increasing loss of land has led to much research in to strategies to minimise the impact of salinity. Secondary dryland salinity is the result of rising saline groundwater, suggesting that its effects may be lessened, stabilised or reversed by managing ground water levels, which are related to many aspects of land management (Kennett-Smith et al., 1994). Typically, methods such as replanting trees, better management of surface water (for example by the construction of drainage systems), and increasing water usage by improving crops/pastures are used to deal with the problem.

For areas which are badly degraded, planting salt-tolerant species which provide food for grazing animals such as sheep may give some productive return, with the added benefit of limiting effects such as soil erosion (Malcolm, 1993). This form of stategy is typically referred to as an *agronomic* strategy.

An important component of the management of the problem is firstly to assess the amount of land that is already saline and secondly to monitor what effect different remedial actions may have. Bore holes provide the most accurate way to measure the depth to ground water, and testing of the water gives an accurate measure of the salinity. These methods can provide detailed monitoring of small sites, but the cost typically precludes detailed monitoring of large areas by this scheme.

For larger-scale assessments of the extent of salinity, aerial photo interpretation has been used in some areas (Nulsen, 1981), but this is time-consuming and expensive and hence not appropriate for monitoring purposes. More recent efforts (Wallace and Wheaton, 1990; Wheaton et al., 1992) have shown that Landsat TM remotely sensed data provide a cost effective means of estimating the extent of salinity at a regional and catchment scale, and hence provide a basis for broad-scale monitoring of the problem.

A limitation of using only Landsat TM data in isolation is that landcovers such as poor pastures have similar spectral characteristics to indicators of salinity such as barley-grass (Wallace and Wheaton, 1990). By incorporating ancillary

data such as soil type information and landform morphology, known relationships relative to salinity can be included in the mapping process to remove some of the obvious errors encountered.

Two separate study areas are considered in this thesis: an area in a 270mm to 420mm rainfall zone, and another in a 500mm-750mm zone. The locations of these areas are shown in figure 3.1.

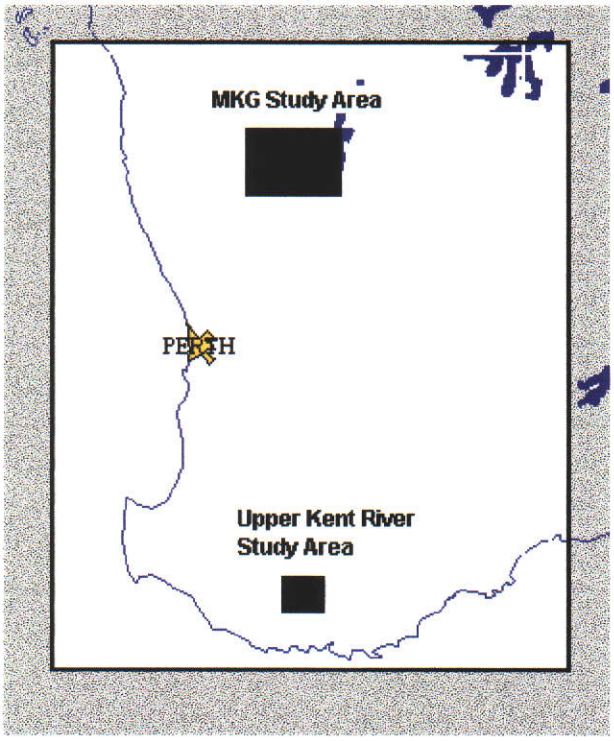



Figure 3.1: Relative location of the two study areas.

In Chapter 6, some experiments in combining data for the purpose of estimating the extent of salinity are given. These preliminary results are based on early work performed in the Moora/Kalannie-Goodlands (MKG) area, introduced in section 3.2, and serve to test and develop the techniques introduced in chapters 4 and 5. Further, some simple considerations of data obtained for this area give valuable insights into the procedures that may be required to produce interpretations across large areas.

Another important aspect of salinity management is not only to know which areas are presently affected but also which areas may be affected if no remedial actions are taken. For example, a widely used strategy for reducing local (saline) ground water levels is to plant trees adjacent to scalded land. This strategy does not always work, as the land where the trees are planted may itself go saline, with or without the newly planted trees. Typically, each tree will cost roughly one dollar to plant. For this reason, knowing which land is likely to go saline appears beneficial, as future plantings can avoid these areas.

In Chapter 7, Bayesian networks are used as a basis for mapping the historical spread of salinity and predicting which areas are at risk of salinity in the future. Predictions are made for the Kent river catchment, which is introduced in section 3.3.

## 3.2 The Moora / Kalannie-Goodlands Area

### 3.2.1 Background

This area is situated approximately 200km. north-east of Perth (refer to figure 3.1) and extends from the Moora area in the south-west of the region to the Kalannie-Goodlands area in the north-east, as depicted in figure 3.2.

The region is approximately 115km. by 115km. in size, representing an area of some 1.3 million hectares. The Moora end of the region typically receives more rainfall, an average of 420mm. annually, than the Kalannie-Goodlands regions, which receives an average of 270mm. annually. The region is predominately used for the growing of cereal grains and the grazing of sheep.

Initial studies on using remotely sensed data to map salinity were located in this region (Wallace and Wheaton, 1990; Wheaton et al., 1992), and based on some of the limitations discovered, ongoing research was planned. The area became the basis for the *Land and Water Resources Research and Development Corporation* (LWRRDC) funded project, *Detecting and Monitoring Changes in Land Condition*

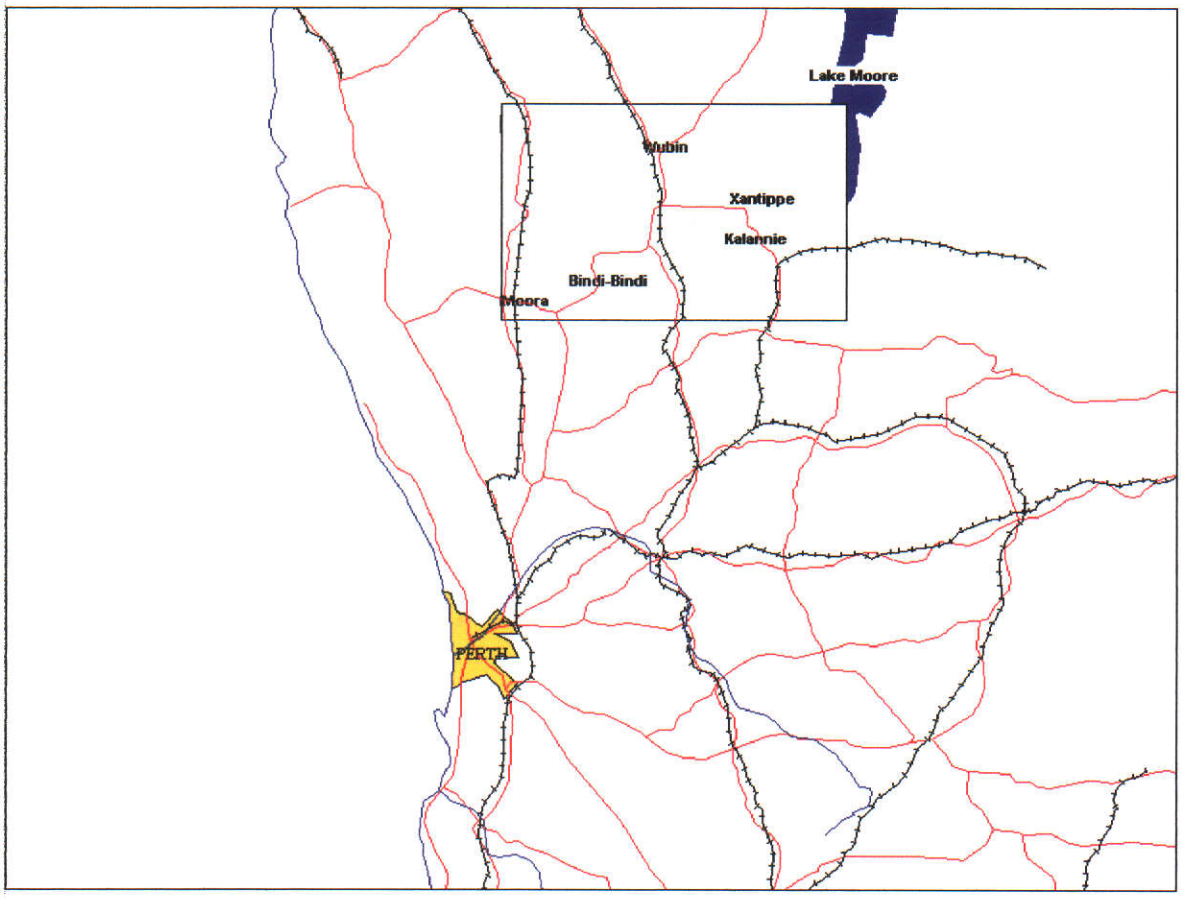



Figure 3.2: The Moora / Kalannie-Goodlands study area.

*Through Time Using Remotely Sensed Data*, which has been subsequently reported on by Furby et al. (1995).

Based on the preliminary work by Wallace and Wheaton (1990); Wheaton et al. (1992), some of the more obvious limitations of using single-date Landsat TM imagery for mapping salt-affected land include: pastures in poor condition along with roads typically are mapped as being salt-affected, leading to over estimation of the estimates of salinity; and the performance of the classifier which was trained in the Moora region degraded significantly towards the north-east of the region considered by Wallace and Wheaton (1990); Wheaton et al. (1992).

Of interest was the use of other data sets that may be able to remove some of the obvious errors encountered. For this reason, a GIS database of ancillary data was constructed. Its contents are described next.

### 3.2.2 The Moora / Kalannie-Goodlands Database

One of the simplest data layers to obtain was road and cadastral boundaries available from the Department of Land Administration (DOLA). The data come in the form of vector polygons which may be converted to raster form to provide simple map masking, for removing roads mapped as saline in the final regional summaries. Note that road dimensions are typically of sub-pixel extent (in relation to Landsat TM data), making it hard to obtain spectral estimates for including a *road* class in an MLC, say.

Simple observation of an average growing season Landsat TM image indicates that the south-western extent behaves differently (spectrally) from the north-eastern extent of the image. For example, figure 3.3 depicts an enhancement of Landsat TM data where bands 2, 3 and 4 are displayed in blue, green and red respectively. Typically with such an enhancement, the redder the image then the greener and more complete the groundcover. Also variations in colour may be observed locally within a paddock.

Based on the following simplistic assumption:

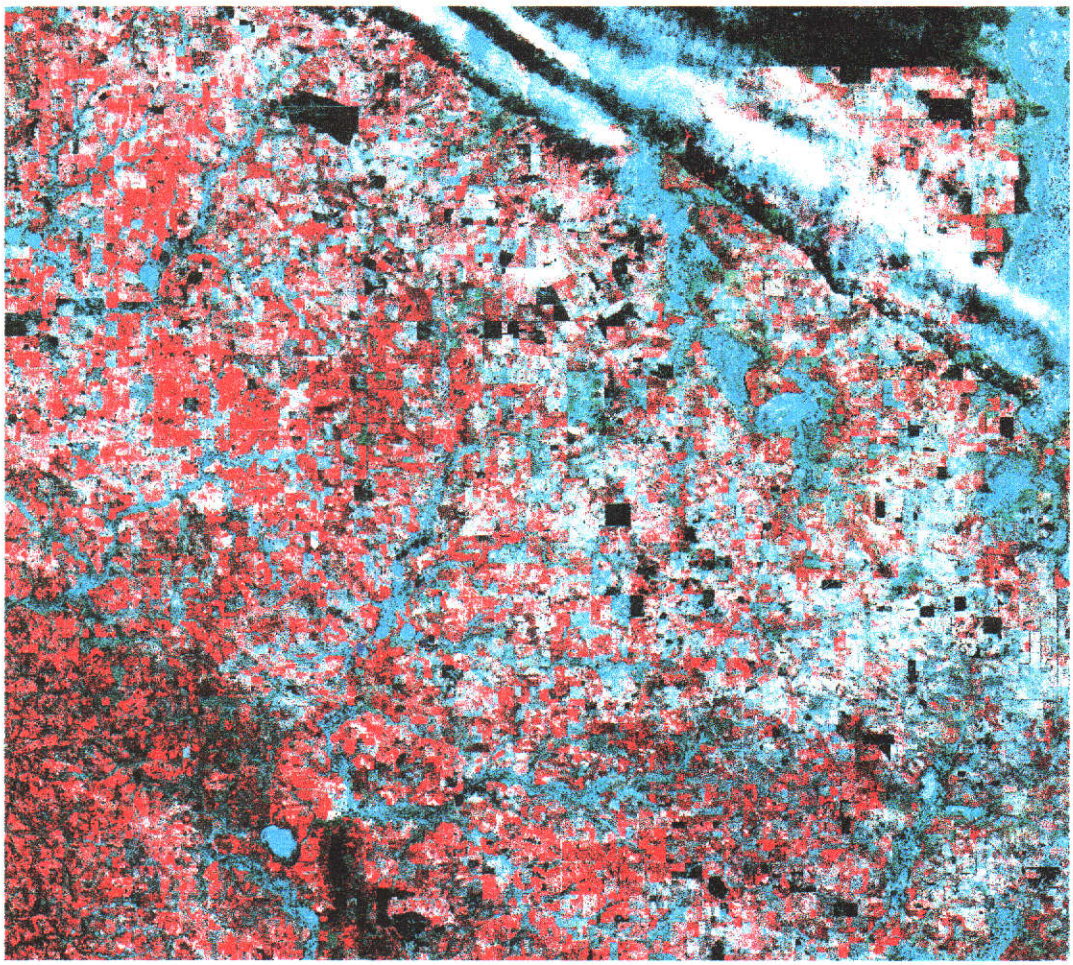


Figure 3.3: Landsat TM enhancement of the MKG study area, with bands 2, 3 and 4 in blue, green and red.

*in lower rainfall areas, lack of adequate rainfall may limit the growth of landcovers, which may also be compounded by the ability of different soils to retain moisture and the amount of moisture an area receives because of effects such as runoff,*

data relating to rainfall, soils and elevation appeared useful and were collected where possible.

Daily rainfall data are collected (via rainfall gauges) by the Bureau of Meteorology at selected locations throughout the wheatbelt. Monthly rainfall figures were obtained from the Bureau for the years 1992 dating back to the 1890s. For each month, these sites were interpolated to form raster images of monthly rainfall. Monthly rainfall images were then combined to form yearly rainfall totals plus an estimate of average yearly rainfall for the period 1973 to 1992. The twenty-year average rainfall image is shown in figure 3.4, along with the sites used to form the image.

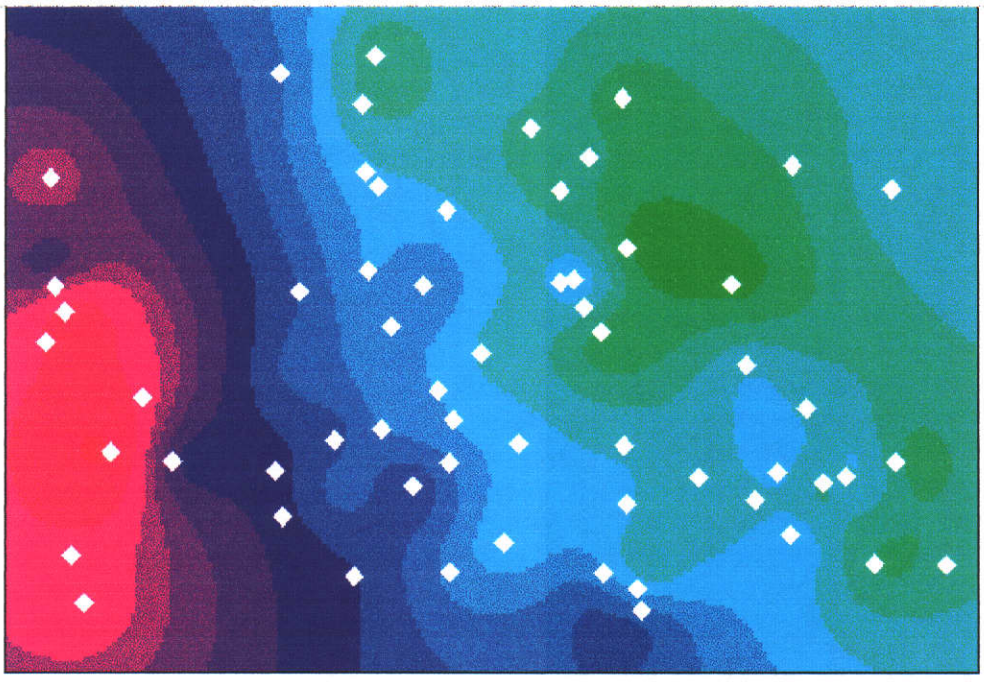


Figure 3.4: Average annual rainfall and associated weather station sites. 420mm (red) through to 270mm (green).

From figure 3.4, we observe the general trend of the redder parts of image 3.3 corresponding to higher average rainfall, which may provide some insights into the trend in the remote sensing data.

Soils data were obtained from local farmers and farming groups in the form of digital vector polygons. Based on the trends in the rainfall data, soil information was gathered for regions in the high and low rainfall regions. Other data available from the farming groups included digitised stream lines and drainage works, but the completeness of the data varied and hence is limited in its usefullness.

Digital elevation data in the form of digitised contours were available for the area and stored in the GIS. Data in the western extent of the study were sampled at increments of 10m in elevation, whereas data toward the eastern extent were sampled at 20m increments in elevation. The interpolated site data are depicted in figure 3.5. In this figure, elevation is depicted by a grey tone, where black represents relatively low-lying land as compared to the white areas which represent relatively high land. Tones in between represent elevations which fall between these extremes.

An accurate set of elevation data was obtained for a small area of the study region so that errors associated with the broad-scale elevation data could be examined. Data for the small area were derived from intensive stereo photo interpretation, giving very accurate relative elevation measures, and differential GPS was used to register, giving high absolute accuracy in elevation. A roughly regular lattice of elevation samples spaced 75m apart resulted, with sub-metre accuracy in elevation being quoted. This work was performed by members of Agriculture W.A. and the Surveying department at Curtin University.

With no means of showing otherwise, the *fine-scale* data were assumed to have negligible errors and were used to examine and estimate likely errors in the broad-scale elevation data, which are of more interest for problems considered here.

Spectral class estimates for use with the MLC were available for the Landsat data considered in this region. The estimates were derived by an expert as part of

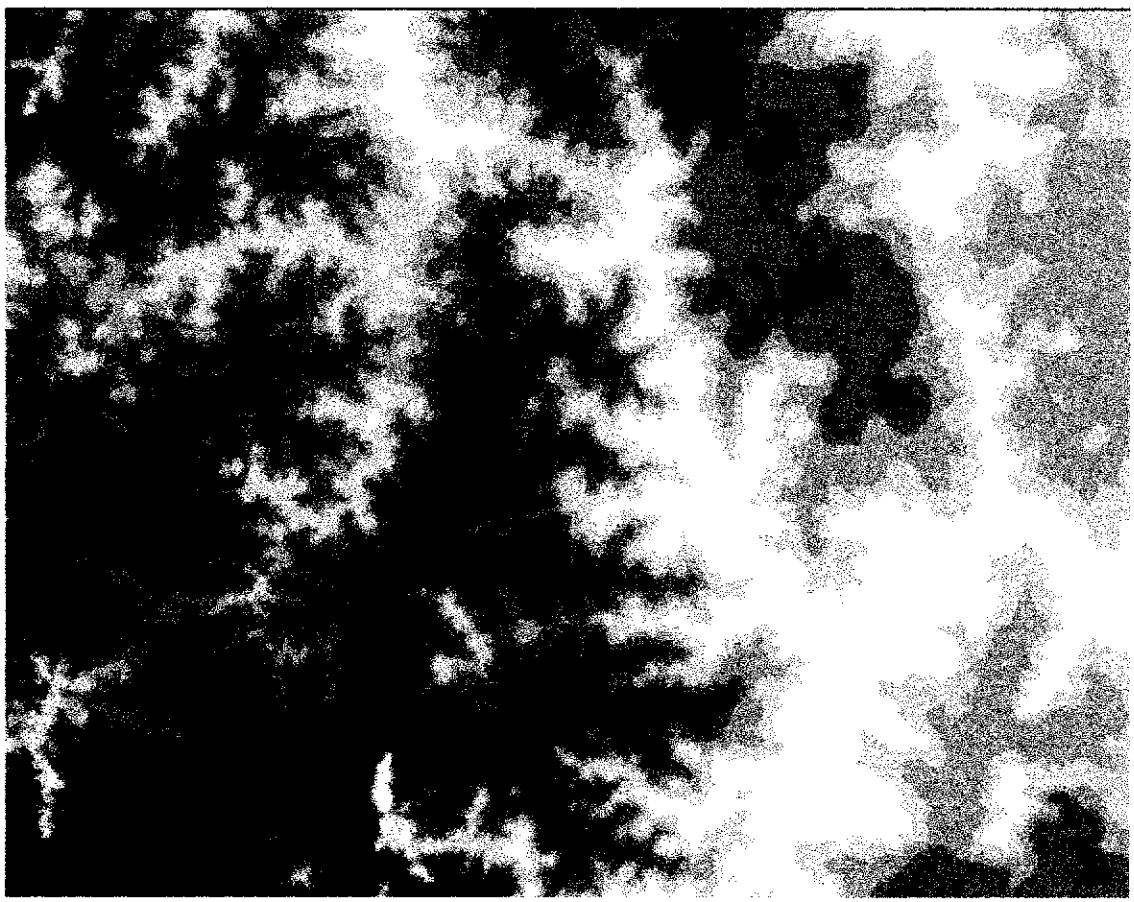


Figure 3.5: Digital elevation model derived from broad-scale elevation data. Black denotes relatively low areas and white denotes relatively high areas.

the *Detecting and Monitoring Salt-affected Land* project (Furby et al., 1995). To validate the mapping accuracy of the MLC, the expert had collected (validation) sites of known land cover/condition. These sites include cropping histories and farm information provided by farmers, and sites recorded from previous studies (Wallace and Wheaton, 1990; Wheaton et al., 1992, 1994).

The land cover classes were grouped into six broad classes:

- Cereal crops - crops including wheat, oats and barley.
- Lupins - lupin crops.
- Pastures - includes any growth in paddocks that are left fallow or are used for grazing sheep. This is perhaps the class with the most variability in plant canopy as it includes sown and self-sown pastures (seeds including those from the previous years crop are left to self germinate) each under different grazing pressures.
- Salt Affected - areas of land considered to be salt-affected. Dead trees, bare scalds, indicator species such as barley grass and salt-tolerant species such as samphire may exist in these areas.
- Remnant Vegetation - vegetation that existed prior to clearing.
- Water - water.

Spectral signatures were derived for the following classes:

1. *GoodCrop* - cereal crops that are considered to be in *good* condition. Typically such crops have a dense canopy cover and vigorous growth.
2. *PoorCrop* - cereal crops that are considered to be in *poor* condition. Typically such crops have a sparse canopy and relatively poor growth.
3. *Lupins*.

4. *GoodPasture* - pasture considered to be in good condition. Typically such pastures have good growth and include sown and ungrazed pastures.

5. *BarePasture* - pasture considered to be in poor condition. Typically this class is composed of grazed pastures.

6. *PastureLikeSalt* - pastures that appear spectrally similar to salt-indicator species such as barley grass, but are not salt-affected.

7. *SaltLikePasture* - salt-affected land that appears spectrally similar to pasture, but is not pasture. Typically this class includes indicator species such as barley grass.

8. *SaltMarginal* - marginally salt-affected land.

9. *SaltSure* - land that is severely salt-affected and typically mapped with greater accuracy than the other classes.

10. *RemVeg* - remnant vegetation.

11. *Water*.

Where convenient, the above classes will be denoted $L = \{l_1, l_2, \ldots, l_{11}\}$ respectively in chapter 6.

## 3.3 The Upper Kent River Catchment

The Upper Kent River Catchment is situated approximately 350km. south-east of Perth and 70km. north-west of Albany (as depicted in figure 3.6 below) in a relatively high rainfall (500-750mm.) region of the state. The land within the catchment is predominatly used to grow pastures for the grazing of animals.

The physiology of the catchment was described by Ferdowsian and Greenham (1992). That report describes the uplifting of the land which diverted the southward flowing ancient rivers west-ward. Later, the Kent river cut through

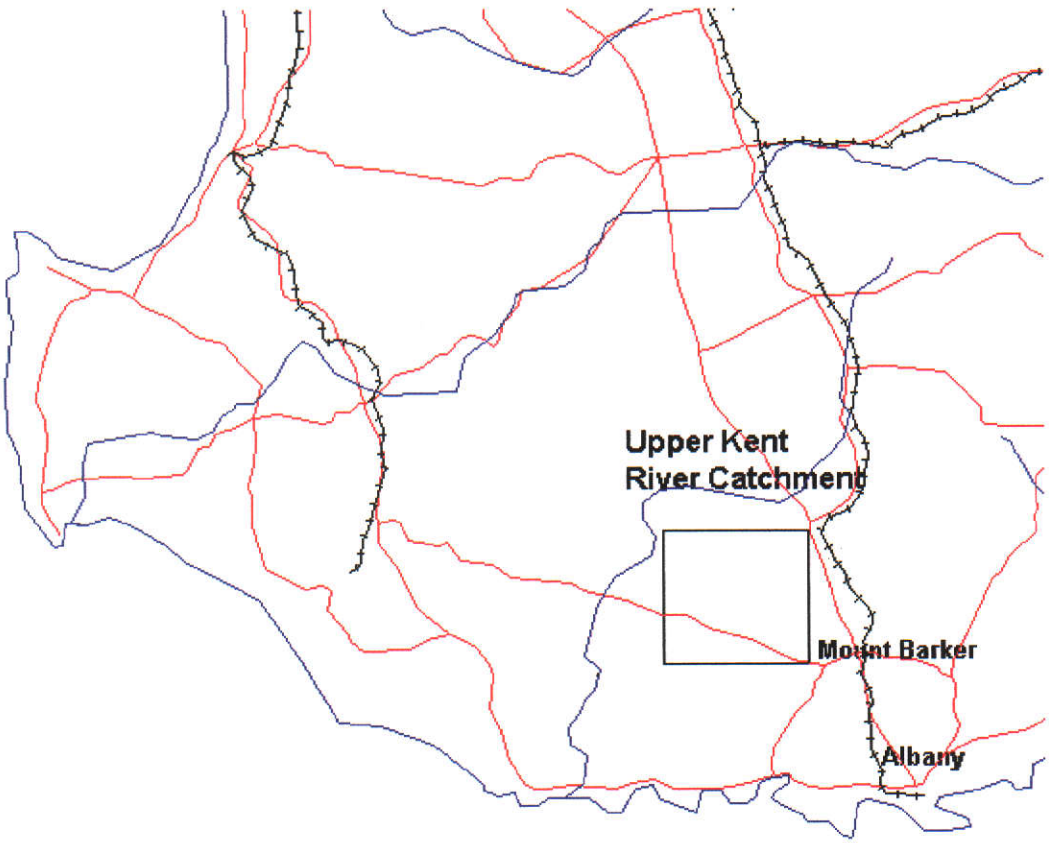



Figure 3.6: Relative locality of the Upper Kent River Catchment.

the uplifted land resulting in the catchment having three distinct hydrogeological zones:

- a zone of catchment divides in the north, east and south;
- a central zone that has stagnant flats, flood plains and playa lakes in the valley floors; and
- a rejuvenated zone in the south-west of the catchment that has active erosional valleys and hence more dissected terrain.

In 1924, Wood explained the increase in soil and stream salinity resulting from the destruction of native vegetation. By 1948, only isolated pockets of land had been cleared and clearing remained slow during the 1950s. Mass clearing occurred in the 1960s, and by 1973 approximately 62% of the catchment had been cleared, which represented most of the land suitable for growing pasture. Clearing activity since 1978 has been small. During the period 1978-1988, a few farmers started planting trees on limited areas to combat salinity problems. Since 1988, some areas have been planted with commercial tree crops, reducing the total area cleared to 58% by 1994.

Because of its relatively high rainfall and relatively recent clearing history, the catchment should be considered atypical of most regions of the West Australian agricultural region, which is predominantly wheat/sheep country.

Unlike lower rainfall regions of the wheatbelt, pastures and remnant vegetation survive in slightly to moderately salt-affected areas, making the mapping of salinity by remotely sensed data difficult. Compounding this, previous research by the CSIRO W.A. Remote Sensing Group has found that salt-affected pasture is less easily mapped using Landsat TM data than other salt-affected land-covers. For this reason, the incorporation of ancillary data appears necessary.

### 3.3.1 The Upper Kent River Catchment Database

The GRASS database was formed as part of the LWRRDC project *Integrating Remotely Sensed Data With Other Spatial Data Sets to Predict Areas at Risk from Salinity*, and is documented in the subsequent report (Evans et al., 1995). The data base contains the following:

1. Landsat MSS data for the date 19/9/1977.
2. Landsat TM data for the following dates: 7/8/1988, 14/9/1990, 1/9/1991, 22/9/1993 and 8/8/1994.
3. Digital elevation data in the form of contour data sampled at 5m. elevation increments.
4. Road and cadastral boundary masks.
5. Salinity ground truth for selected areas, as provided by an expert and subsequently scanned in and rectified.

The Landsat TM data were classified using the MLC (Evans, 1995b) and the results also stored in the GIS.

Formation of the DEM and extraction of features from the DEM is considered in section 5.

## 3.4 Summary

In the West Australian wheat-belt, the clearing of land for agricultural production has resulted in rising saline ground water, leading to the loss of previously productive land to salinity. Traditional methods for assessing the extent of salinity include farmer-based surveys and the manual interpretation of aerial photographs. The first method is inaccurate and tends to underestimate the extent. Also the location of the salinity is not recorded, only its extent expressed as a percentage.

The second method is time consuming and therefore relatively expensive. For these reasons they are not well suited for providing future updated information on the extent of the problem.

Wallace and Wheaton (1990) demonstrated that Landsat TM data could be used in the Moora region to (spectrally) discriminate, and therefore map, many landcovers not associated with salinity and certain types of landcovers associated with salinity. However, some classes remained spectrally inseparable and some errors in the resulting classification were made. Further, the mapping accuracies deteriorated when extended eastward toward the lower rainfall catchment of Xantippe.

Based on the relative success of the above work, and its limitations, the use of multiple sources of Landsat TM data as well as other ancillary data provided a likely means of reducing mapping errors, and became the focus of future research efforts, some of which are presented in Chapter 6.

Firstly, the MKG ancillary data base described in section 3.2.2 was formed, and represents the types of data available for broad-scale monitoring. Data such as road vectors are useful for identifying and masking out roads which are typically spectrally confused with salinity. Broad-scale rainfall data are useful for identifying relatively homogeneous regions of rainfall and therefore a gross estimate of which areas would be expected to have similar crop growth/management strategies[1]. Also demonstrated is the lack of complete coverage of potentially useful data such as soil type.

Another important problem is not only to know the current extent of salinity, but also where it is likely to spread to in the future. This has farm management implications, such as avoiding the construction of new dams in areas that will be affected by salinity in the future. Salinity prediction is considered for the Upper Kent River Catchment in Chapter 7 (the area was introduced in Section 3.3).

[1]The author notes that other factors such as geology and regional soil type are also considered important.

# Chapter 4

# Implementation of algorithms relating to Bayesian networks

## 4.1 Introduction

Bayesian networks provide a means of combining data in the presence of uncertainty, and have been used in domains including medicine, agricultural science and more recently, to a limited extent, in image analysis. One advantage of the probability propagation technique considered here is the ability to model multi-variable interactions, as opposed to simple two-way interactions of earlier systems such as used by Shortliffe and Buchanan (1975); Adams (1976); Gordon and Shortliffe (1985); Pearl (1986a); Shenoy and Shafer (1986). A consequence of the generality of the approach is that computational demands may increase as compared with the earlier propagation schemes. Perhaps for this reason, other authors have not considered this general approach for combining large sets of data typical in remote sensing and GIS applications, although with each new generation of computers increasing in power, this may not be the case in the future.

In this chapter, the implementation of the general probability propagation scheme is described, as well as new techniques specific to the combining of remotely sensed and GIS data. Algorithms which aim to improve computational efficiency

are developed and experiments to assess the likely improvements are described. Also, a new way of incorporating images that have uncertainty explicity specified is presented. This approach makes it possible to combine data which explicitly have uncertainty modelled by other means with the rules expressed by a Bayesian network. For example, data positional uncertainty as modelled by Kiiveri (1996) produces for each data class a probability of class membership: that is the data which need to be combined are probability images. These data may be combined with Bayesian networks using the approach to be described.

This idea in part leads to the idea of using an *expectation maximisation* algorithm, or EM algorithm for short, for model parameter (rule strength) estimation. The implementation of the EM algorthm is described.

More specifically, section 4.2 describes the implementation of the probability propagation algorithm described by Lauritzen and Spiegelhalter (1988). From an algorithmic viewpoint, the technique has two main components: (1) the *initialisation* component, which takes the DAG and associated probability tables and forms structures necessary for probability propagation; and (2) the *propagation* component, which performs the actual calculations. Algorithms in the initialisation component are only performed once, whereas the propagation algorithms are performed many times. For image processing, the propagation component is performed once for each pixel.

As noted by Kjaerulff (1990), the most important step in constructing an efficient framework for probability propagation is to achieve a good triangulation during initialisation. In general, this problem is NP-hard (Chickering et al., 1994; Arnborg et al., 1987) and is typically solved using heuristic algorithms (Kjaerulff, 1990). For the applications considered in this thesis, which typically require the propagation algorithm to be applied millions of times, an algorithm which optimally solves the triangulation problem is desirable. A branch-and-bound algorithm which achieves this is described in section 4.3.

Another aspect of using Bayesian networks is the problem of estimating pa-

rameters. In section 4.5 the EM algorithm is reviewed and its implementation described. To the author's knowledge, the EM algorithm has not previously been considered for general Bayesian networks used in the remote sensing/GIS domain.

## 4.2 Algorithm Implementation

The Bayesian network code used for the experiments in this thesis includes an implementation of the algorithm of Lauritzen and Spiegelhalter (1988) with added features useful for combining image processing techniques with GIS data.

The code was implemented on a Unix based, SUN workstation. Code was written in the programming language C.

The Bayesian network probability propagation algorithm can be loosely divided into two components: algorithms relating to initialisation (as discussed in section 2.3.4); and those relating to the propagation of probabilities (as discussed in section 2.3.3). For the initialisation component, which will be referred to as `Initialisation()` in the following, apart from the work considered in section 4.3, implementation followed the scheme described in section 2.3.4. For the propagation component, which will be refered to as `Propagate()` in the following, algorithm 2.26 was implemented. The structure is shown in figure 4.1.

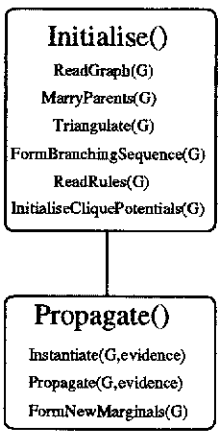



Figure 4.1: Standard Algorithm.

For image processing, the implementation explicitly distinguishes between two

forms of evidence (data): *hard* evidence, and *virtual* evidence. Entering hard evidence amounts to forcing a variable to have a particular state, and is the most common way of entering evidence in applications of Bayesian networks in other fields. Entering the evidence, *variable $B$ has state $b_k$*, is an example of hard evidence.

Earlier work by Pearl (1986a) suggested that evidence may also be entered in the form of *virtual* evidence, although I am not familiar with any other authors adopting this approach in applications using Bayesian networks. Virtual evidence is presented to the system in the form of a probability (or potential). That is, evidence is presented in the form *variable $B$ has state $\psi_e^B$*. This allows probabilistic models such as those which incorporate uncertainty in maps (for example see (Kiiveri, 1996)) to be executed independently and then combined with other data with the probability propagation code. Hard evidence amounts to specifying the state of a variable (the state of the variable is known with absolute certainty).

In the following discussion the subscript $e$ will denote potentials relating to evidence. If one recognises that data may be entered as probability distributions, then using this approach unifies the process of uncertain reasoning with uncertain data. Modelling data uncertainty is a field in its own right.

Clearly, hard evidence is a special case of virtual evidence, but has the following important distinction: hard evidence for variable $B$, say, forces $B$ to have a particular state, whereas virtual evidence treats $B$ as a latent variable and hence the final belief in $B$ may not be equal to the evidence $\psi_e^B$, but is based on the combination of all evidence.

This feature of virtual evidence allows uncertainty in the data, as distinct from uncertainty in the rules applied to the data, to be modelled separately and then incorporated into the data-combination process. For example, MLC typically produces images and resulting classification label uncertainty images which may be entered as virtual evidence to the data combination scheme, as opposed to the more standard practice of entering the class labels as hard evidence.

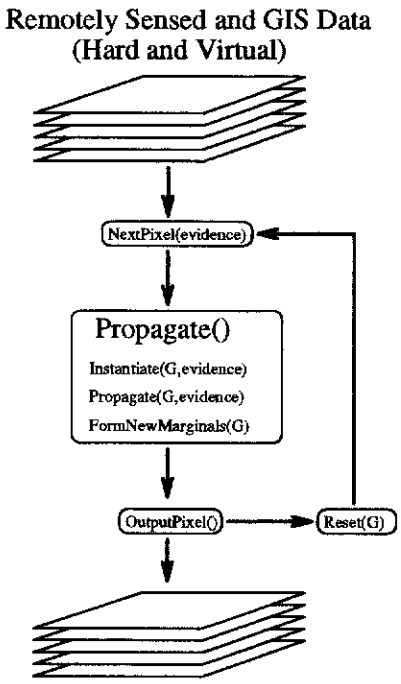



Figure 4.2: Standard data processing scheme.

### Incorporating virtual evidence

Let $T = (V, \Delta)$ represent the hypertree obtained from a Bayesian network, and $\psi_i$ the potential parameterisation for clique $\delta_i \in \Delta$. Next, let $\psi_e^B$ denote evidence about variable $B$. Then, if $B \in \delta_j$, evidence is entered to produce the conditioned potential $\psi_c$ as follows:

$$\psi_c = \psi_j \otimes (\psi_e^{B \to \psi_j}). \tag{4.1}$$

The process of entering evidence will be referred to by `Instantiate()` in the following.

As implemented, the code will allow any number of (possible virtual) layers to be entered as evidence, after which the data are combined, and then maps of the (probabilities of) states of any number of variables are produced.

The incorporation of virtual evidence naturally leads to the idea of using an expectation maximisation (EM) algorithm (Dempster et al., 1977) for estimating system parameters. Normally, conditional probability tables are estimated by

simply counting the number of times a particular configuration is encountered. Given virtual evidence configurations may never be explicitly observed but only partially observed. The EM algorithm, in part, uses these partial counts to form parameter estimates. The EM algorithm and its implementation is described in section 4.5. Firstly, the problem of triangulation is considered as it forms the basis of efficient propagation.

## 4.3 Optimal Triangulation

Triangulation is used for devising a decomposition of variables for propagation of probabilities in Bayesian networks (Lauritzen and Spiegelhalter, 1988), with the basic requirement to be satisfied being that the variables can be arranged in a tree of cliques. This section considers the problem of finding an optimal triangulation of an undirected, vertex-weighted graph. A comparison of heuristic and optimal algorithms has been previously considered by Kjaerulff (1990). In testing the optimal algorithm, Kjaerulff (1990) compared the heuristic solutions with the best, possibly non-optimal, solution obtained by the optimal algorithm after a maximum of two hours of computation; this was reasonable considering the relatively large graphs that were used in his experiments.

For remote sensing and GIS applications, the graphs will tend to be much smaller than those considered by Kjaerulff (1990), but the volume of data, and hence the number of propagations required, will be relatively large. For this scenario, obtaining an optimal solution may be worth the extra effort. For this reason, we restrict ourselves to fairly small graphs and re-examine how far from optimal the best of the heuristics, *minimum weight*, tested by Kjaerulff (1990) actually is.

In particular, a branch-and-bound algorithm is described and compared with the *minimum weight* heuristic considered by Kjaerulff (1990). Before describing the algorithm, some necessary graph theory terminology is introduced.

### 4.3.1 Terminology

Firstly we assume an undirected graph $G = (V, E)$.

An undirected graph is said to *triangulated* if every simple cycle of length four or more possesses a chord.

If a graph is not triangulated, then it requires edges to be added to the graph so that the resulting graph is triangulated. Hence the process of triangulation may be accomplished by a procedure which detects simple chordless cycles of length four or more and then adds the appropriate edges. This is the strategy used here.

Given a Bayesian network, obtaining a good triangulation is the most important step to obtaining a computationally efficient decomposition of variables (Kjaerulff, 1990).

A Bayesian network associates with each vertex $v_i \in V$, a discrete (random) variable having $w_i$ ($w_i > 1$) states. In this discusion, *size* relates to the state space that needs to be represented by the network. For example, if $A$, $B$ and $C$ are variables having 2, 3 and 5 states respectively, say, then the table $\psi(A, B, C)$ would have $size = 2 \times 3 \times 5 = 30$.

For probability propagation in a Bayesian network, each (maximal) clique of the underlying undirected graph must be parameterised. The size of a clique is given by the product of the number of states of its associated variables. That is, if $\{v_1, v_2, \ldots, v_k\}$ represents the set of vertices in clique $\delta$, then the size of the clique is given by

$$size(\delta) = \prod_{i=1,2,\ldots,k} w_i. \tag{4.2}$$

Thus if the graph decomposes into the set of cliques $\Delta = \{\delta_1, \delta_2, \ldots, \delta_m\}$, then the size of the graph is given by

$$size(G) = \sum_{i=1,2,\ldots,m} size(\delta_i). \tag{4.3}$$

The computational efficiency for such graphs may be dominated by the size of

the largest clique in the triangulated graph (Lauritzen and Spiegelhalter, 1988), or as shown by Jensen and Andersen (1990), directly related to the total size of the network. That is, having a graph of many small cliques is preferable to having one or a few large cliques.
The problem to solve is:

> *Given a graph* $G = (V, E)$, *where vertex* $v_i$ *has weight* $w_i$, $w_i > 1$, *triangulate* $G$ *to form a graph* $G_T$ *such that* $size(G_T)$ *is as small as possible.*

We notice that by adding edges to a graph, the number of vertices in a clique will increase and since $w_i > 1$, we have an increasing cost function, which suggests this problem can be solved by branch-and-bound methods.

In the following discussion, we assume basic familiarity with branch-and-bound methods (Lawler et al., 1985). In brief, the branch-and-bound method for an optimisation problem involves the decomposition of the given problem into a number of smaller sized sub-problems. The important components of this procedure are branching, bounding and searching strategies. The branching strategy dictates the manner in which a given problem is decomposed into sub-problems. For the problem considered here, the sub-problems consist of finding a triangulation of a graph $G$ with the added requirement that some edges are included and some are excluded from the solution. A solution to a relaxation of the sub-problem provides a lower bound on the value of the objective function. Here the relaxed sub-problems are graphs with the triangulation constraint dropped. All sub-problems which are triangulated are called *feasible solutions.* During the search, a feasible solution with the lowest cost is considered to be the *best* feasible solution.

Bounding compares the objective function of a sub-problem with that of the best current known feasible solution. If the cost of a sub-problem is greater than the current best feasible solution, then this sub-problem, and all sub-problems that may be generated from it, need not be considered. In this way, bounding restricts

the number of sub-problems generated and hence may result in a computationally efficient algorithm.

Searching strategies define in which order sub-problems are considered. In this application, a depth-first search strategy was used.

In the next section, the branch-and-bound algorithm is described.

### 4.3.2 The Algorithm

It is assumed that from a given Bayesian network we have generated the undirected graph $G_0 = (V, E_0)$ in the normal way (marrying parents and dropping edge directions), where vertex $v_i \in V$ has weight $w_i$. Assuming $G_\# = (V, E_\#)$ represents the complete graph, then for a graph $G_k$ we maintain the set $I_k \subset E_\#$ of *included* edges and the set $X_k \subset E_\#$ of *excluded* edges. Initially, $I_0 = E_0$ and $X_0 = \emptyset$.

Assuming $G_k = (V, I_k)$ has a simple cycle $C = \{v_1, v_2, \ldots, v_n, v_1\}$, $n \geq 4$, and $D = \{(v_{n-1}, v_2), (v_i, v_{i+2})\}$ for $i = 1, 2, \ldots, n-2$ represents the set of chords for $C$, then two new sub-problems $G_{k+1} = (V, E_{k+1})$ and $G_{k+2} = (V, E_{k+2})$ are generated as follows:

1. INCLUDE the edge:

   $E_{k+1} = E_k \cup e;\ \ e \in D;\ \ e \notin X_k;$

   $I_{k+1} = E_{k+1}$

   $X_{k+1} = X_k$

2. EXCLUDE the edge

   $X_{k+2} = X_k \cup e; e \in D; e \notin I_k;$

   $E_{k+2} = E_k$

   $I_{k+2} = I_k$

Included edges cannot subsequently be removed and excluded edges cannot subsequently be added. Including an edge works toward a feasible solution. Excluding an edge has the effect of forcing the problem towards infeasibility.

The algorithm starts with $G_0$, looks for a violating cycle and if one is found the sub-problems $G_1$ and $G_2$ are generated and placed on a stack. At this stage, the problem $G_0$ needs to be considered no further and is removed from the stack. The problem $G_2$ is then taken off the stack to generate (if they exist) sub-problems $G_3$ and $G_4$. This process continues until all sub-problems have been considered.

If a feasible solution (no violating cycles exist) or an infeasible solution (no edges can be added to the cycle) is generated, then no further sub-problems are generated from these two cases. In the case of a feasible solution, the solution weight is checked against the current best solution weight, and if it is less this becomes the best current solution. Sub-problems with weight greater than the best current solution are rejected, and hence no further sub-problems are generated from these cases. Initially, the heuristic cost (Kjaerulff, 1990) is used as the upper-bound. In this way, all possible solutions are considered and hence the optimal solution obtained.

The algorithm relys on the following operations:

- A modified shortest path algorithm is used to detect a violating cycle.
- Given such a cycle, an edge $e \in D$ needs to be considered.
- Given a sub-problem, all maximal cliques need to be extracted, using an algorithm such as described by Bron and Kerbosch (1973), and the cost of the sub-problem is calculated.

A simple chordless cycle is detected by the following algorithm:

- For a vertex $v_i \in V$, extract the neighbour set $Ne(v_i)$.
- For each pair of vertices $v_j, v_k$; $j \neq k$; $v_j, v_k \in Ne(v_i)$, use a shortest-path algorithm to find a simple path $\{v_j, \ldots, v_k\}$ of length 2 or more from $v_j$ to $v_k$ (if one exists).
- If such a path exists, then this path plus vertex $v_i$ constitutes the violating cycle $C = \{v_i, v_j, \ldots, v_k, v_i\}$.

The algorithm branches on the included edge sub-problem first, resulting in feasible solutions being found quickly, thus enabling the search space to be efficiently pruned.

### 4.3.3 Results

The algorithm was tested on cycles of various orders. Cycles were chosen, as they have many possible triangulations. For example, a cycle having $|V| = n$ requires that a minimum of $n - 3$ edges be added to be triangulated. These edges must be chosen from a possible $(n - 3)n/2$ edges, giving a total number of possible sub-problems of the order of ${}^{(n-3)n/2}C_{n-3}$, a potentially large number.

As an example, fifty problems were generated for each order, where the vertex weights were randomly generated from the interval [2,20]. The problems were solved on a SUN SPARC10 computer. The performance statistics are summarised in tables 4.1 and 4.2, where:

- $|V|$ is the number of vertices in the cycle.
- $MinP$ $(MaxP)$ is the minimum (maximum) number of sub-problems considered for the solution of any one problem.
- $MinT$, $MaxT$ and $AvT$ is the minimum, maximum and average time in seconds required to solve any one problem.
- $AvOptSize$ and $AvHeurSize$ is the average size of the optimal and heuristic solution respectively.
- $NumImp$ is the total number of problems not solved optimally by the heuristic.
- $AvImp$ is the average improvement, expressed as a percent, obtained by using the optimal algorithm as compared with the heuristic solution.

| $\|V\|$ | MinP | MaxP | MinT(s) | MaxT(s) | AvT (s) |
|---|---|---|---|---|---|
| 10(50) | 3333 | 15983 | 2.00 | 9.00 | 4.8 |
| 15(50) | 272697 | 5115497 | 256.0 | 5146.0 | 1748.5 |

Table 4.1: Algorithm performance - number of sub-problems generated and algorithm processing times.

| $\|V\|$ | AvOptSize | AvHeurSize | NumImp | AvImp (%) | MaxImp (%) |
|---|---|---|---|---|---|
| 10(50) | 839.7 | 873.0 | 28 | 4 | 21 |
| 15(50) | 1220.8 | 1290.7 | 41 | 5 | 22 |

Table 4.2: Algorithm performance - solution sizes.

- *MaxImp* is, for each class of 50 problems, the maximum reduction in the heuristic size obtained by the optimal algorithm.

From the tables, it is observed that of the 100 graphs tested, 69% were not optimally triangulated by the heuristic solution. However, on average the heuristic solution was within about 5% of the exact solution, which is a good result from the heuristics point of view. The largest improvement observed was about a 22% reduction in graph size, which for intensively used networks may be worth pursuing.

The heuristic algorithm took negligible time (less than 0.01 seconds) to solve each of the problems considered. The solution times recorded for the optimal algorithm suggest that it is feasible to optimally triangulate graphs for applications considered in this thesis, which typically would not have cycles much larger than 15 vertices. In general it would be expected that applications would have graphs with many leaf nodes and hence larger graphs could be solved optimally in reasonable time. However, as the order of the graph increased, a large increase in the number of sub-problems needed to be considered was observed, which reflects the increasingly intractible nature of this problem as the size and number of cycles within a graph grows.

Given a Bayesian network, triangulation is only ever performed once, whereas

for image processing, the triangulated Bayesian network may be applied millions of times which may warrant the extra cost of finding an exact solution.

## 4.4 Processing Data that Contain No Attribute Uncertainty

For the development of complex models operating on large data sets, processing time may become a limitation. In this section, a new scheme that provides computational advantages is conceived. Here we assume that no layers representing virtual data are present, that is, all evidence is *hard* evidence. Although it may be desirable to use *virtual* evidence, this approach may be used to reduce the time required to test different model prototypes, and then, once a model is selected, process the (virtual) data in the standard way to produce the final product.

### The Processing Scheme

The scheme makes use of the idea that as particular configurations of variables are combined by the network, the results are stored and recalled the next time the same configuration is encountered. Computational advantages will be achieved if the time associated in storing and recalling the results is less than performing a network propagation.

Typically, because of the spatial relationships between adjacent pixels associated with remotely sensed and GIS data, many pixels will have the same configuration and hence many network computations will be saved.

For cases where the data have a small number of possible configurations, an obvious approach is to simply precompute the configurations and store the results in a look-up table (LUT). That is, for each node for which results are desired (which for the present discusion will be called $TargetNodes$), simply precompute (by artificially instantiating the network with all possible configurations and then

performing a propagation) the LUT $P(TargetNodes|DataNodes)$ and then process the data by looking up the results from the LUT.

A potential problem with this approach is that if many data layers are involved, then the total number of possible configurations may be unmanageable (in terms of storage space) or larger than the dimensions of the data sets being processed. Further, some particular configurations may simply not exist in the data for a particular region.

Given these problems, a scheme that only computes configurations actually found within a particular data set is desirable, and having computed them once, they need to be stored and retrieved from a (potentially) sparse LUT. A binary search tree meets the storage requirements.

### Binary search trees

Standard Unix routines (tsearch(), tdelete()) provide a means for building, accessing and deleting binary search trees, and these functions were used in the implementation described here.

Conceptually, accessing data in a binary search tree is relatively straight forward: starting from the tree root, descend down the tree until the desired node is found, branching to the *left* if the current node is *greater* than the objective node and branching to the *right* if the current node is *less* than the objective node. If the objective node is not found, insert it in its relevant place. If there are $N$ nodes in the search tree, access time is of the order $\log_2 N$.

### The Proposed Processing Scheme

The processing scheme devised performs as follows: for a given pixel, check (using the binary tree) whether the configuration has been previously processed and retrieve the results; else process the data using the network and store the newly processed configuration in the binary tree. If the configuration was previously in the search tree, then this will be called a *hit*, otherwise it will be called a *miss*.

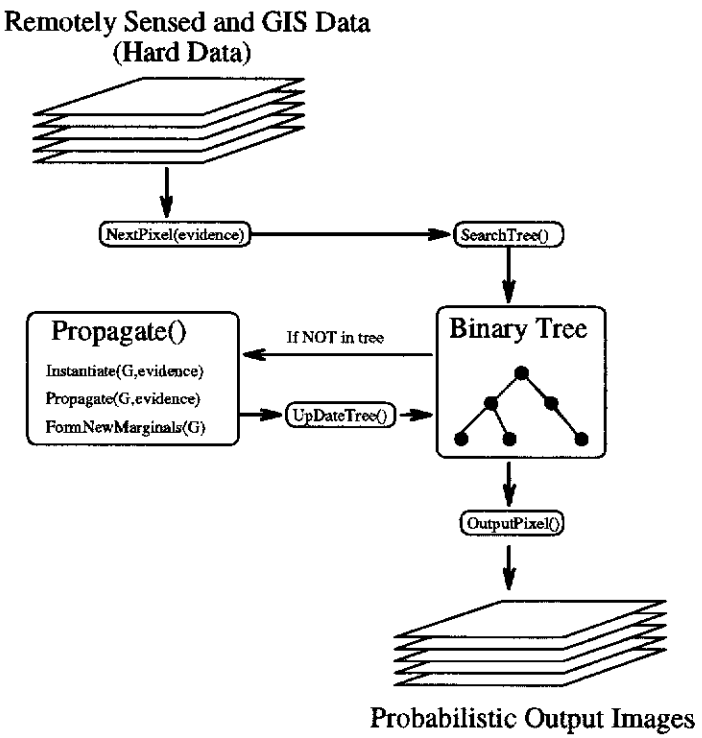



Figure 4.3: Incorporating the binary search tree.

This algorithm is depicted in figure 4.3.

For complex data sets, there is the potential that many different configurations exist and that the binary tree may become large, requiring much storage space and relatively long access times, perhaps longer than the time required for a network propagation. If the time taken to access the binary search tree and to perform a network propagation are denoted as $TreeTime$ and $NetTime$ respectively, then the total time required to process a pixel is given by:

$$TotalTime = TreeTime + (miss)NetTime$$

That is, every time a *hit* occurs, a network propagation is saved, but every time a *miss* occurs, then the time taken is the binary tree access time plus the network propagation time.

To avoid the potential situation that $TreeTime >> NetTime$, the following strategy is employed:

1. The first line of data is processed to estimate the $NetTime$ for a single line of data

2. For subsequent lines, the binary tree is incorporated and the $TotalTime$ for processing each line calculated

3. When $TreeTime$ becomes greater than $NetTime$, then the binary tree is deleted and the process of building the tree starts again.

**Comments**: A variety of tree-size maintainance schemes could be employed, for example *delete all configurations that have not been used within the last* 50 *lines*, say, or *delete all least accessed configurations*. As configured, testing typically produced hit/miss ratios of 90% or more. In the worst case scenario, every pixel has a unique data vector, processing times increase as the tree is maintained but never produce a hit. As would be expected, the efficiency of the approach varied depending upon the Bayesian network specified and the data being processed. For example, combining multiple classifications produced by Landsat TM tends to produce large search trees, as these classifications typically do not have the spatial pattern associated with, say, a soil map. To date, the worst-case performance of the approach was when applied to prototyping the network in chapter 7. The approach reduced processing time to approximately one third of the time needed by the Bayesian network used in isolation, which is a useful improvement.

## 4.5 The EM Algorithm and its Implementation

In this section the implementation of an EM algorithm for Bayesian networks is described.

In the following discusion, we assume a Bayesian network $G(V, E)$, parameterised by the conditional probability tables $P_1(v_1|u_1), P_2(v_2|u_2), \ldots, P_r(v_r|u_r)$, $v_k \subseteq V$, $u_k \subseteq V$, $k = 1, \ldots, r$. From the network and tables we obtained the hypertree $T = (V, \Delta)$ and potentials $\psi$ for each clique $\delta \in \Delta$.

If the *true* labels were observed then the maximum likelihood estimate of $P_k(v_k|u_k)$, $k = 1, \ldots, r$, would be

$$\hat{P}_k(v_k|u_k) = n(v_k, u_k)/n(u_k), \tag{4.4}$$

where $n(v_k, u_k)$ represents the number of times each possible combination of states is observed for the variables in the sets $v_k$ and $u_k$. The quantity $n(v_k)$ is similarly defined. That is, $\hat{P}_k(v_k|u_k)$ is the relative frequency of observing each combination of variable states. In the following, this approach will be called the *obvious* approach.

If some (or all) of the variables are unobserved, the counts $n(v_k, u_k)$ and $n(v_k)$ are unknown. Instead the expected counts are estimated given a provisional estimate of $P_k(v_k|u_k)$. On the basis of these expected counts a new estimate for $P_k(v_k|u_k)$ can be obtained. This iterative process, known as an EM algorithm (Dempster et al., 1977), is performed until the parameters have converged in some sense.

Suppose at the $q$'th iteration of the EM algorithm the estimate of $P_k(v_k|u_k)$ is $P_k^{(q)}(v_k|u_k)$.

The *E-step* consists of calculating the expected counts

$$n^{(q)}(v_k, u_k) = \sum_{\forall \text{ pixels}} m^{(q)}(v_k, u_k) \tag{4.5}$$

where $m^{(q)}(v_k, u_k)$, $k = 1, \ldots, r$ is the conditioned marginal distribution obtained from $\psi^A$, $A \in \Delta$, $\{v_k \cup u_k\} \subseteq A$. The *M-step* consists of maximizing the likelihood on the basis of the expected counts, that is to calculate

$$P_k^{(q+1)}(v_k|u_k) = n^{(q)}(v_k, u_k)/n^{(q)}(u_k). \tag{4.6}$$

In effect, the conditional probabilities formed by the EM algorithm are an average of all partial configurations observed.

The absolute maximum difference in $P_k^{(q)}(v_k|u_k)$ and $P_k^{(q+1)}(v_k|u_k)$ is used to gauge the convergence properties of the EM algorithm. That is, if $MaxDev$ signifies the convergence measure, this corresponds to calculating:

$$MaxDev = max|P_k^{(q)}(v_k|u_k) - P_k^{(q+1)}(v_k|u_k)| \; \forall \text{possible state combinations of } v_k, u_k. \tag{4.7}$$

Heuristically, the parameters may be considered to have converged when $MaxDev$ is less than some user-specified threshold.

### Properties of the EM Algorithm

The EM algorithm has the following appealling properties:

- The degree to which the configuration of a number of variables is known is considered when estimating the unknown parameters. For instance, in some circumstances, the configuration may be definitive, that is the only possible configuration. In other instances, the evidence may indicate that there may be two or more possible configurations, with different configurations more probable than others.
- The algorithm is relatively simple to implement.

Some of the less appealing properties are:

- The algorithm may take many iterations to converge, which in the context of RS/GIS means many processing passes of the data.
- In general, there is no guarantee that the model will converge to the global maximum.
- In some instances, parameters may not be identifiable given the data used for training.

Expert rules provide one possible means for limiting the impact of the negative aspects of the algorithm.

### The Use of Expert Rules

Rules from experts may be used to define the initial parameter estimates prior to commencement of the EM iterations. Care needs to be taken to ensure that the observed data contain sufficient information to estimate all the required parameters. Expert rules can help by establishing that some events are not possible, some events are adequately known, or where the necessary observation data are (perhaps temporarily) missing, can be used to subjectively define some parameters.

For this reason, the ability to be able to update a subset of parameters was included in the implementation. As implemented, any subset of the rules specified by a conditional probability table can be preserved and the remaining rules updated using the EM algorithm. For example, consider the table $P(A|B,C)$, where parameters have been fixed for all but the following rule:

$$\textit{If } B = b \textit{ and } C = c \textit{ then } A \textit{ with } P(A|b,c).$$

In this case, the E-step of the EM algorithm is unchanged, but during the M-step only the parameters $P(A|b,c)$ are updated.

## 4.6 Summary

This chapter has described various aspects of the implementation of the probability propagation code used for the experiments in this thesis.

From an implementation perspective, the algorithm can be considered as being composed of the following two main components:

1. `Initialise()`: The routines which must be executed before any updating on the basis of evidence can be performed.

2. `Propagate()`: The routines which combine the evidence to form updated beliefs. That is, those routines which are executed for each data vector entered.

One of the most important stages in the `Initialise()` component is to obtain a good triangulation, which determines the computational aspects of the network as used in the *Propagate* component. This problem, which in general is *NP-hard*, has been considered in some detail by Kjaerulff (1990), who compared a number of known heuristic solutions with an approach based on simulated annealing. Testing of a branch-and-bound algorithm was mentioned, but no details of the algorithm were given. In comparing the heuristic and exact solution methods, Kjaerulff (1990) set a time limit for exact methods and hence heuristic solutions were obtained. No idea of the true optimality of the heuristic solutions was given. For the large volumes of data typical with remote sensing and GIS, the intention of revisiting this problem was to examine whether it is worth making the extra effort to solve the triangulation problem exactly, or to establish how good the heuristics actually are, and thus save more time in the future when large data sets may be routinely processed.

To achieve this, a branch-and-bound algorithm was implemented and the (exact) results compared with that obtained from the best of the heuristics as tested by Kjaerulff (1990). For the graphs tested, 31% of the problems were solved optimally by the heuristic. On average, they were within approximately 5% of being optimal, a good result from the heuristics perspective. The largest deviation observed was a network which was 22% larger than the optimal result, which demonstrates that some significant improvements are possible by solving the triangulation problem exactly.

The Bayesian network implementation considered here allows two types of evidence to be entered, *hard* and *virtual* evidence. Virtual evidence is presented to the system in the form of a probability distribution.

An approach for improving processing times during the `Propagate()` stage was then proposed. The approach assumes that all evidence is hard evidence and is suitable for many problems and particulary useful during prototyping of a model. The approach makes the assumption that data vectors will tend to repeat

themselves in GIS data, because this form of data tends to define homogeneous regions much larger than the size of a pixel. In this way, the results of previously processed data vectors are stored and recalled when subsequently encountered. When not all data are hard data, then an approximation may be made by assuming they are hard data and using the above approach. This allows for quicker network prototyping.

Finally, noting that extracting good probabilistic assessments from experts may be difficult, an estimation technique known as Expectation Maximisation has been implemented. This approach is used for estimating the conditional probability tables and assumes that:

1. The structure of the network is known.

2. Data for some variables may not exist (or be incomplete) and their states must be inferred from other data.

The approach may be used in an automated way or used to refine parameter estimates provided by experts.

# Chapter 5

# Experiments in Landform-Morphology Feature Extraction

## 5.1 Introduction

This chapter describes experiments in stratifying digital terrain models into landform units, and in particular landform units applicable to mapping and monitoring in the W.A. agricultural region. An added emphasis is placed on mapping and monitoring salinity.

Typically, accurate elevation data are not available on a broad scale: broad-scale elevation data are available in the form of contour data sampled at elevation increments of 5, 10 or 20 metres. Error is likely to exist in the resulting DEM, which in turn is likely to affect the reliability of the feature extraction algorithms.

To gain understanding of the errors likely to be encountered in the broad-scale DEM, *accurate* (see section 5.2.1) elevation data are obtained for a small region within the study area and the resulting *accurate* DEM is compared with the broad-scale DEM (see section 5.2). Observation of the *accurate* DEM, and the landform features of interest, are used in a pragmatic way to define the parameters best

suited to form the broad-scale DEM. Experiments in partitioning the DEM are then performed.

Firstly, existing landform partitioning techniques are tried and some serious limitations are encountered. Many of the published algorithms performed poorly as they are not suited to the very flat nature of the terrain in question. New approaches were needed and their development and performance is described.

Understanding the hydrogeological characteristics of a region is important in determining the occurance of salinisation (Ferdowsian, 1994; Ferdowsian and Greenham, 1992). Broad-scale algorithms that to some extent model hydrology appear advantageous. Algorithms based on drainage network extraction provide a starting point. Intuitively, such a model would be expected to separate valleys from hilltops and ridges by virtue of the fact that little water accumulates on the latter features. Also, as salinity is the result of local water table levels, it would be expected that water accumulation would act as a predictor of salinity.

In section 5.3, existing water-accumulation algorithms are applied and compared. As will be seen, the single-flow and multiple-flow algorithms give vastly different results even in non-flat regions. Both have serious limitations in extremely flat regions. The effect these limitations have on published approaches is examined.

In section 5.4, algorithms that explicitly attempt to extract flat features are considered. An existing *flat area classifier* is implemented and tested on the DEM. Based on its limitations, a new flat area classifier is devised.

In section 5.5, the idea of combining flat regions with water accumulation algorithms is extended further by considering flat features as flow objects, resulting in a new water accumulation algorithm which combines the results of section 5.4 and section 5.3. This new algorithm proved relatively successful on a small-scale and performed well on large-scale data.

## 5.2 The Digital Elevation Model

Two sources of elevation data are available for regions contained within the MKG study area:

1. a detailed set of sites accurately measured for a small region; and

2. a set of sites derived from contour data for the MKG study area.

### 5.2.1 DEM Construction

#### The *Accurate* DEM

The accurate data are derived from photo-interpretation, which provides point estimates of elevation on a roughly regular grid having 75m spacing, as depicted in figure 5.1. For a given location, the mean and variance of the distance to the nearest data point is 22m and 156m respectively. The greatest absolute distance is 103m. This is relatively dense in comparison with the broad-scale data, which has a mean and variance of 119m and 10094m respectively, and a greatest absolute distance of 1800m.

A raster DEM having a 30m pixel resolution was desirable, as this coincides with the resolution of Landsat TM data with which it is eventually to be combined.

Because the data points were relatively dense, the data were interpolated using a minimum distance interpolation routine (GRASS routine **r.surf.idw**) to produce the DEM which is depicted in figure 5.2. In this figure, elevation is coloured yellow representing the lowest parts of the terrain through green, blue, purple and then red which depicts the highest part of the terrain.

In the following, it is assumed that this DEM represents the *true* elevation of the region, from which comparisons with the broad-scale data are made. Although the accurate DEM would be likely to also contain some element of error, it is assumed that the errors are negligible in comparison with any errors in the broad-scale data.

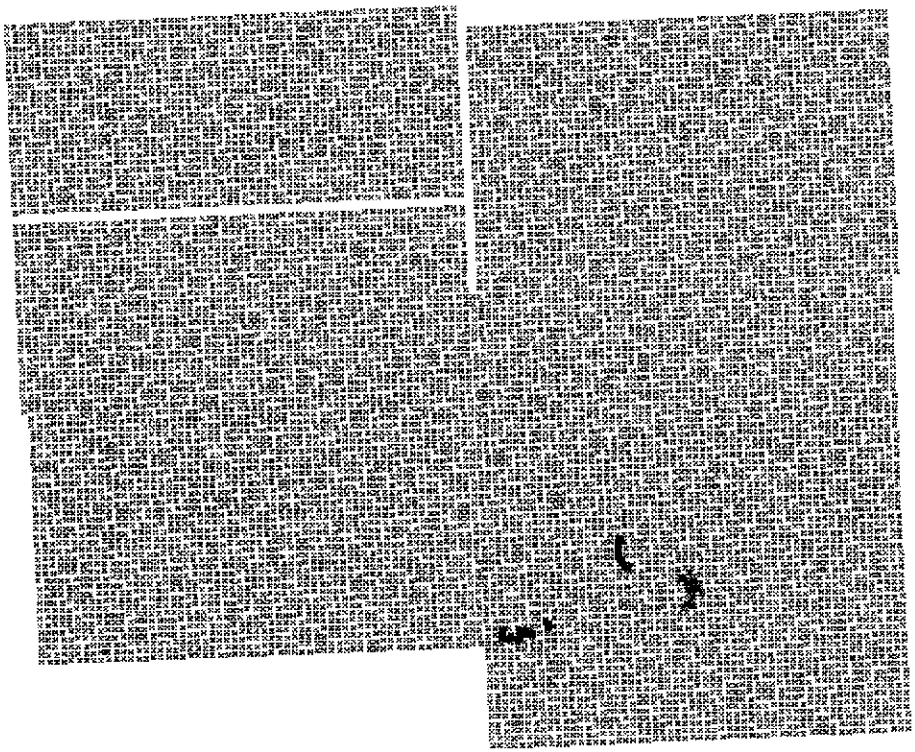


Figure 5.1: Sample sites for the accurate elevation data (area approx. 4km. by 4km.).

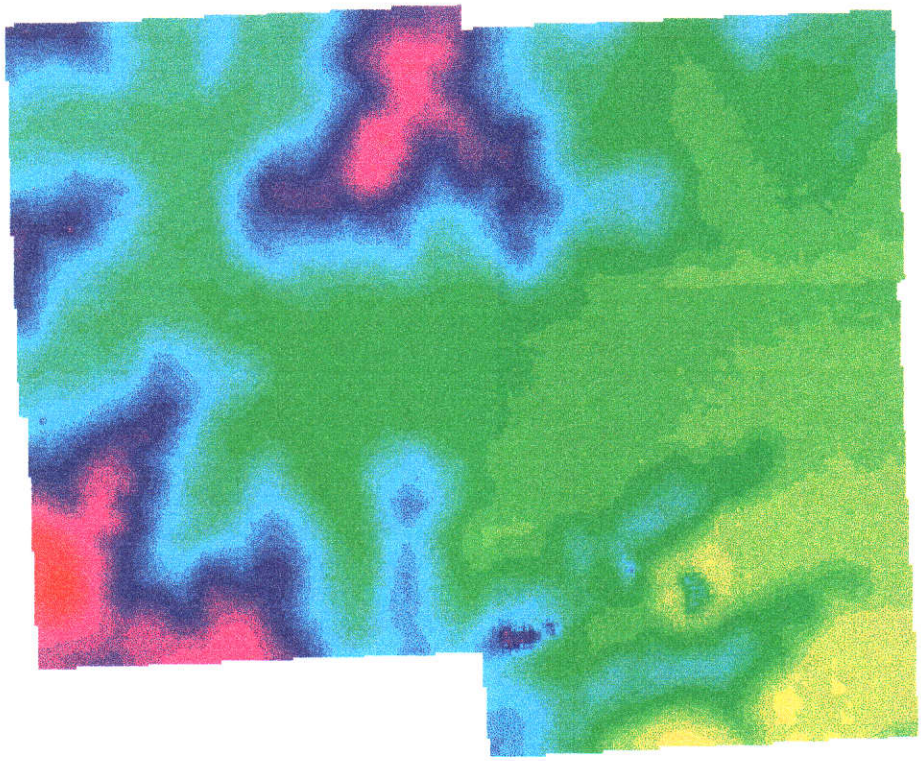


Figure 5.2: DEM derived from accurate elevation data. The colours range from yellow, depicting the areas with the lowest elevation, through green, blue, purple, pink and then red, which depicts the regions with the greatest elevation.

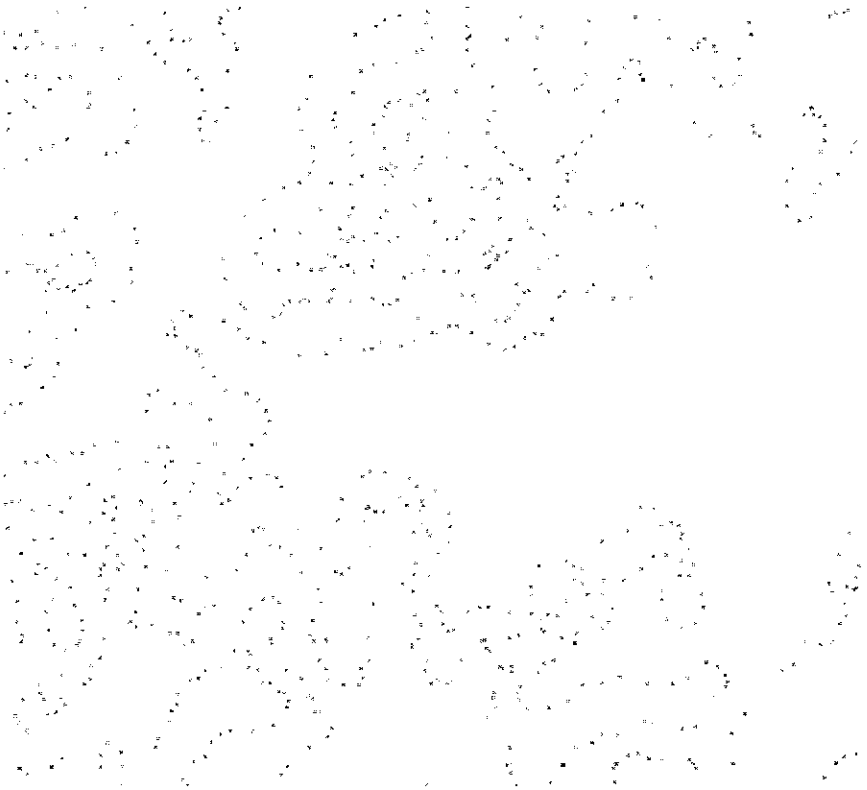


Figure 5.3: Sample sites for broad-scale elevation data.

A characteristic of the region in question is the relatively broad, flat nature of the drainage systems, which tend to go saline. Maintaining this property in constructing the broadscale DEM is desirable.

### The Broad-Scale DEM

Because of the sparsness of the broad-scale data, cubic spline interpolation (GRASS routine **s.surf.tps**) was used to form gridded data layers of elevation, slope, aspect and curvatures[1] A 30m grid was chosen, as this coincides with the resolution of Landsat TM data.

Figure 5.3 depicts the elevation data and figure 5.4 the resulting DEM. Here the colours of figure 5.4 are directly comparable to those in figure 5.2.

[1]For more information on this interpolation proceedure, refer to the GRASS user manual.

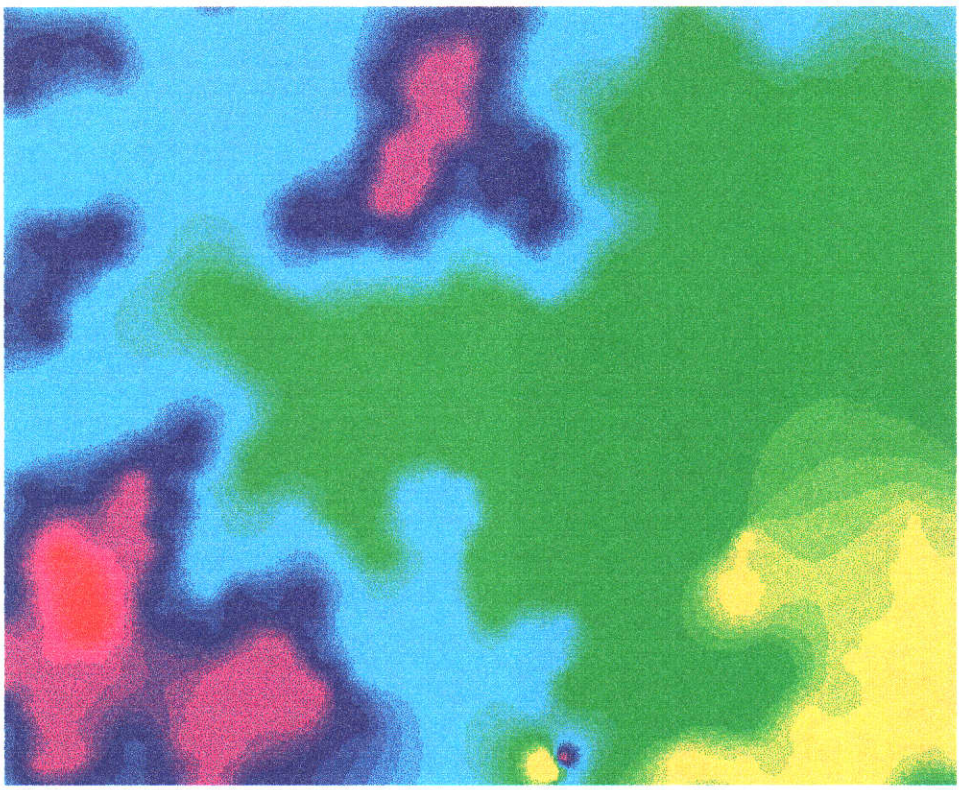


Figure 5.4: DEM derived from broad-scale sites. The colours range from yellow, depicting the areas with the lowest elevation, through green, blue, purple, pink and then red, which depicts the regions with the greatest elevation.

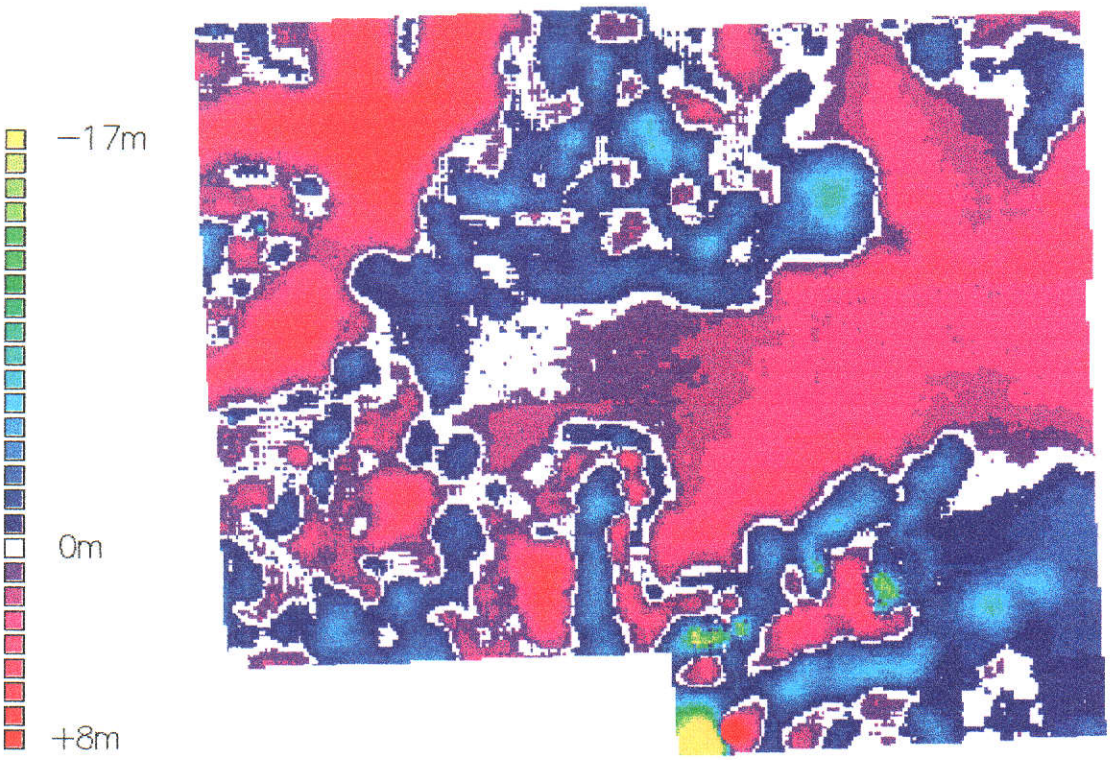



Figure 5.5: Difference image of the DEM derived from the broad-scale data and the *true* elevation.

### 5.2.2 Errors in the broad-scale DEM

The difference in heights of the accurate DEM and broad-scale DEM were calculated on a per-pixel basis, giving a maximum absolute error of 17m and an average absolute error of 2.15m. The maximum errors range from 7m (in the valleys where the true valley floor is lower than the broad-scale DEM) to -17m where there is a breakaway. Figures 5.5 and 5.6 provide a graphical depiction of the errors[2]. In figure 5.5, the colour ranges from red, indicating large positive errors, through to blue, green and then yellow which indicates large negative errors. Where the figure is white, the two DEMs agree. From this figure it is observed that the greatest errors occur in valley floors and on hilltops, which is probably the result of sampling quantization errors. From the three dimensional view of the region given in

[2]The three dimensional drapes used in this section refer to the top left quadrant of the accurate DEM. This was done to avoid cluttered looking views while still conveying the intended point.

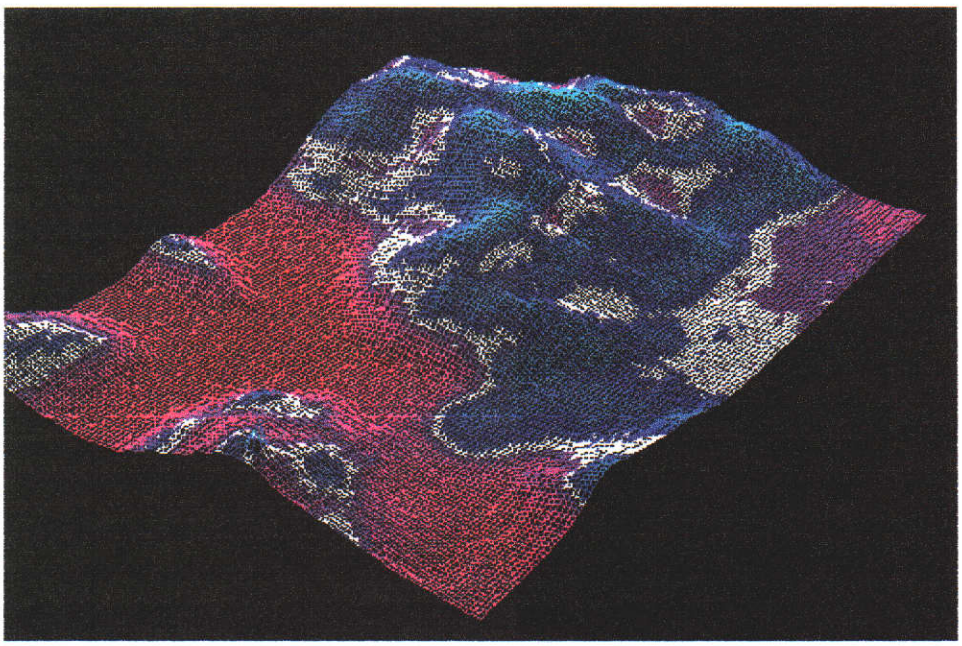


Figure 5.6: Three dimensional perspective of the top left hand quadrant of the image in figure 5.5

figure 5.6 it is observed that the elevation of valley floors is over estimated by the broad-scale DEM and the elevation of the hilltops/ridges is under estimated.

To examine the effect of quantization errors, figure 5.7 was constructed by extracting contour lines from the accurate DEM. The contour lines have 2m intervals, with intervals divisible by ten highlighted in a different colour. The 10m contour lines represent elevation heights at which the broad-scale data were sampled. From the figure, the effect of the 10m sampling interval demonstrates that significant changes in elevation on hilltops and in broad-valleys are ignored, and given the interpolation process of the broad-scale data, result in the heights of these features being under estimated and over estimated respectively. Perhaps more importantly, the boundaries of the broad-valley, as defined by the last lowest contour line before the true valley, have a large lateral displacement from the *true* boundary; the extent of the broad-valley system is consistently overestimated by this definition.

With these points in mind, various existing and newly developed landform

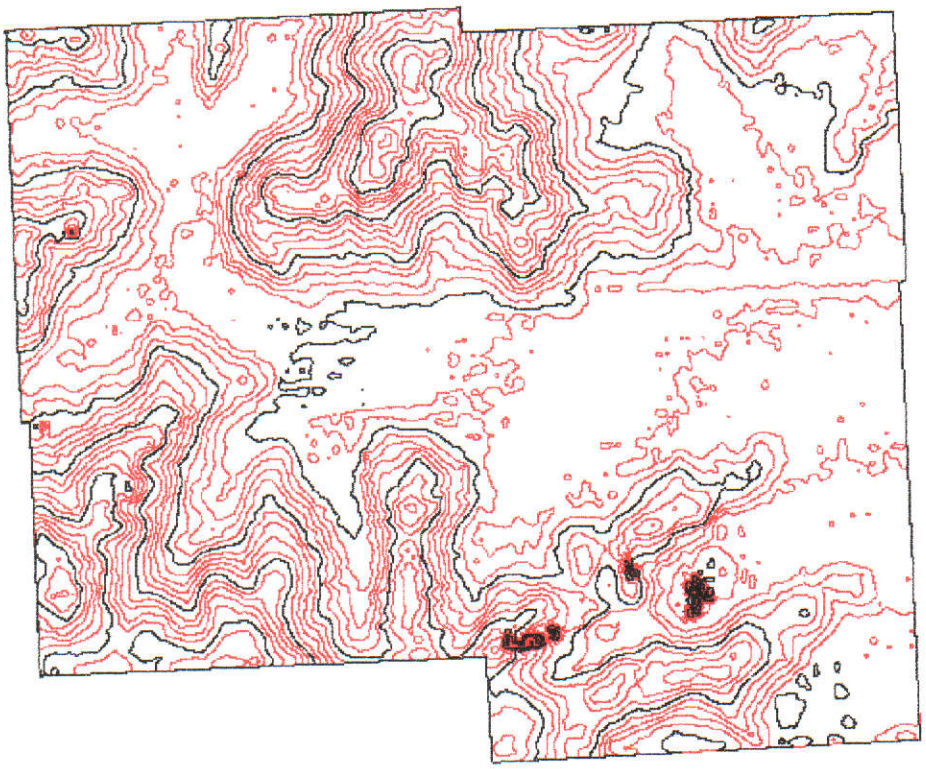


Figure 5.7: Contour intervals extracted from the *true* elevation. The broad-scale elevation data is sampled at elevations depicted by the black contour lines, which when compared with the two metre contour lines shown in red, demonstrates that elevations may be under estimated for hilltops and ridges and over estimated for lower lying areas such as valleys.

partitioning algorithms were applied to the broad-scale data, which is the topic of the next section.

## 5.3 Some Initial Partitioning Experiments

### 5.3.1 Application of Existing Techniques

#### Approach 1: The Statistical Approach

The DEM interpolation process provides five variables which could be used for partitioning. The variables of main interest are elevation, slope and the curvature variables mean, tangent and profile. Intuitively, given the elevation and its first and second derivatives, it should be possible to extract landform features using these variables (for example hilltops (high elevation, low slope and negative second derivatives) and valleys (low elevation, low slope and positive second derivatives)). Because of the spatial extent of the valley systems, it is common that the elevation of a valley in the upper part of a catchment is greater than a hilltop in a lower part of the catchment. For this reason absolute elevation is not considered as a variable in the following landform partitioning experiments.

Based on the above points, training areas for the classes HillTop, Ridges and UpperSlopes, UpperValleys, LowerValleys and BroadValleys were defined by manual examination of the data and used to form a MLC of the data into *landform* units. The variables used were the first (slope) and second (curvatures) derivatives of the DEM. An example of the resulting classification is shown in figure 5.8, where in the figure, the colours red, blue, cyan, green and yellow represent the classes BroadValleys, LowerValleys, UpperValleys, Ridges/UpperSlopes and Hilltops respectively.

The following were observed:

- Large flat regions including hilltops, ridges and valleys had no meaningful curvature measurements. This resulted in these flat regions all being classi-

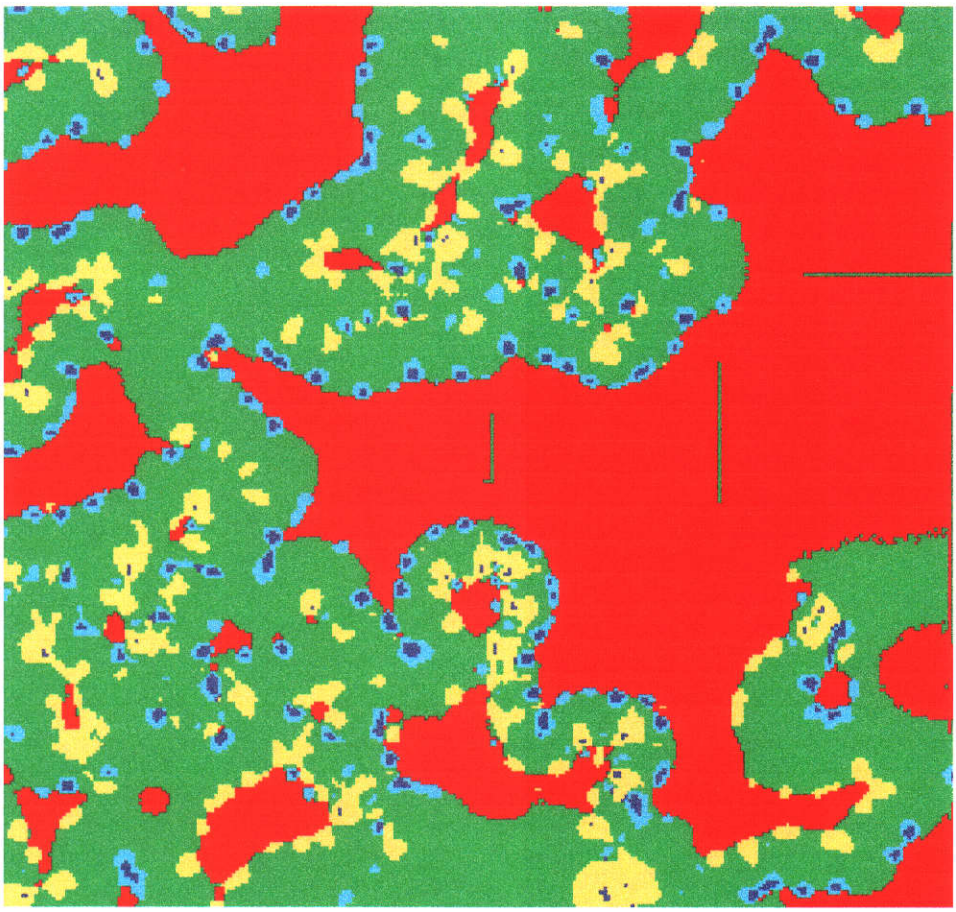


Figure 5.8: Landform Partitioning using MLC of Slope and Curvature Variables.

fied as BroadValley, which is the most likely class for flat features (the area of BroadValleys is typically many times that of features such as Hilltops).

- The LowerValleys and UpperValleys identified by the classification do not realistically reflect the true structure of a drainage system.

#### Approach 2: Extraction of Skeletal Drainage Networks

Much of the salinity in the W.A. wheatbelt appears in the bottom of valley systems, and hence extracting these features is desirable. Many GIS packages have drainage network extraction procedures and may be routinely applied to DEM data. The data were processed with the GRASS routine **r.watershed**, which is based on a single-flow water accumulation algorithm, to derive a drainage network. An example of the network is given in figure 5.9. In this figure, ridge lines extracted by the algorithm have also been included to illustrate the poor performance of these algorithms as was discussed previously.

For this region, the drainage network identified by the accumulation algorithm is typically within 600m of the true drainage network. This error is not suprising given the terrain in question and the DEM used.

For tasks such as salinity mapping, a skeletal model for the drainage system is of limited use as the entire broad-valley system tends to go saline as opposed to only the *defined* drainage channel being affected. An algorithm which better represents the spatial characteristics of the broad-valley system would therefore be beneficial. In the next section a new algorithm that attempts to expand the skeletal network to have the dimensions of the true network is considered.

### 5.3.2 Height-Constrained Valley Extrapolation

Given the skeletal network, it would seem that the spatial extents of the valley system could be extracted by performing a region growing operation that is constrained by height. That is, iteratively expand the region such that a pixel is

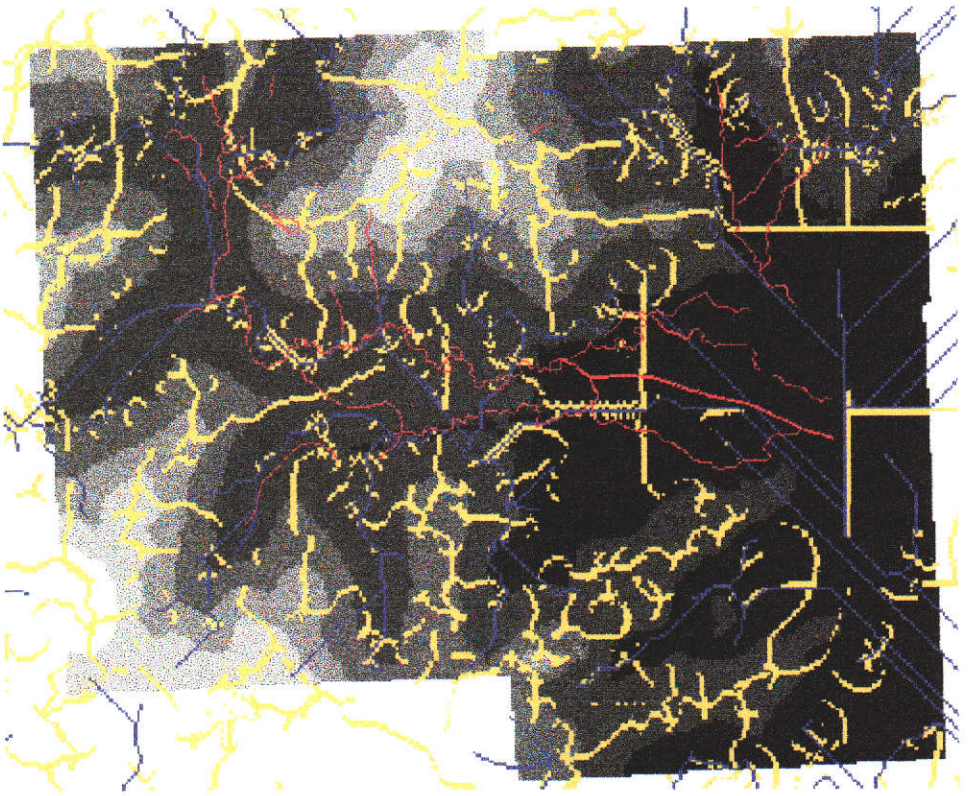


Figure 5.9: Drainage network, in blue, resulting from the GRASS routine r.watershed. The true drainage network is shown in red while the GRASS derived watershed boundaries are shown in yellow.

only included in the region if it is less than, say, $H$m above the skeletal drainage network. A problem with this definition is that the drainage system extracted does not have constant height, so the height above the *nearest* drainage pixel is appropriate. A simple new algorthim, referred to as `HeightAbove`, based on these ideas was conceived and is described below.

Firstly, we note that a feature of distance transformation algorithms such as those described by Borgefors (1984, 1986) is that they essentially perform a *shortest path* calculation of each non-feature pixel to any of the feature pixels. That is, each non-feature pixel ends up having a value equal to the (approximate Euclidean) distance to the *closest* feature pixel.

Although distance transformations do not explicitly identify the nearest feature pixel, a simple modification to the algorithm will allow this (refer to the pseudo code in the appendix). Once the nearest feature pixel is identified, then it is a trivial task to obtain its elevation and hence the height that any pixel is above the nearest feature pixel.

The new algorithm makes use of two temporary images, a *distance* image $D$ and a *labels* image $L$, and requires the user to input a *feature* image $F$ which contains the features for which we are calculating the height above. The algorithm proceeds as follows:

1. Initialise the distance image to value 0 where there is a feature pixel, or a large number (*infinity*) otherwise.

2. For each feature pixel, generate a unique number and initialise the corresponding label image pixel to this value.

3. Perform a modified distance computation as given by the pseudo code (see Appendix A).

   The modified distance computation performs two passes of the image, propagating distances away from the feature of interest, resulting in an image which represents the distance of a location from the nearest feature pixel.

With the `HeightAbove` implementation, the feature label is also propagated, allowing the nearest feature pixel to be identified.

On completion of this operation, each pixel in the label image contains the label of the nearest feature pixel.

4. For each feature pixel, retrieve its height from the DEM and store the label and height in a look-up table.

5. Generate a "height above" image by subtracting the height of the nearest feature pixel from the DEM. The label image and look-up table is used for this purpose.

The computations require two passes of the data to propagate the labels, and then one more pass to form the final *height above* image. Further, the entire computation never requires more than two lines of data to be stored in RAM at any one time.

The above algorithm was applied, using the broad-scale DEM and the drainage network generated using the GRASS *r.watershed* algorithm, to obtain a *"height above drainage network"* image; examples of the results are given in figure 5.10 and 5.11. From figures 5.10 and 5.11 it appears that the "height above" categories may provide a partitioning into upper, middle and lower landform positions which would be useful for salinity mapping.

To test the usefulness of the partitioning, the "height above" variable was compared to salinity information.

The *height above* image ($H$) was stratified into eight classes and and compared with salinity information, as summarised in tables 5.1 and 5.2.

From tables 5.1 and 5.2, we observe that most salt falls in the class 0-1m above the skeletal drainage network. For this class, 58% of the pixels are saline. From the tables, it is apparent that this variable is better at indicating where salinity is not rather than where it is. For instance, no pixels in the >4m class are saline. This is partially the result of the DEM not defining the valley floors well, and as

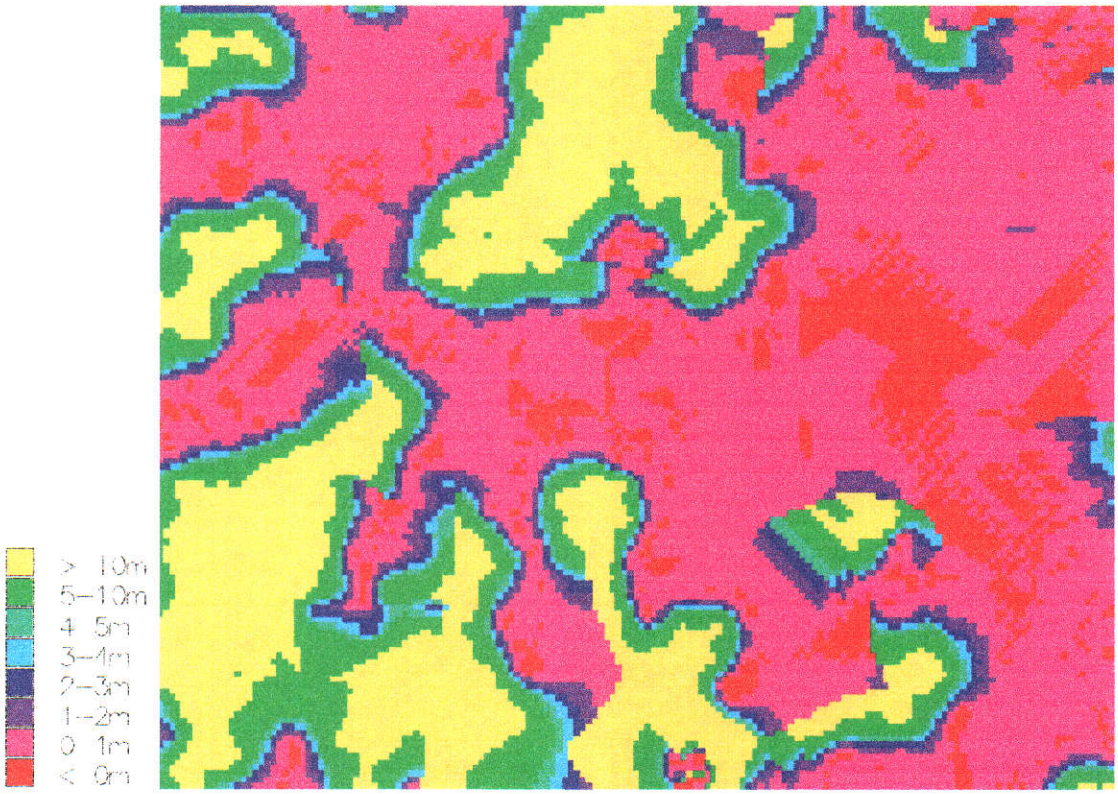



Figure 5.10: DEM partitioning using `HeightAbove()`.

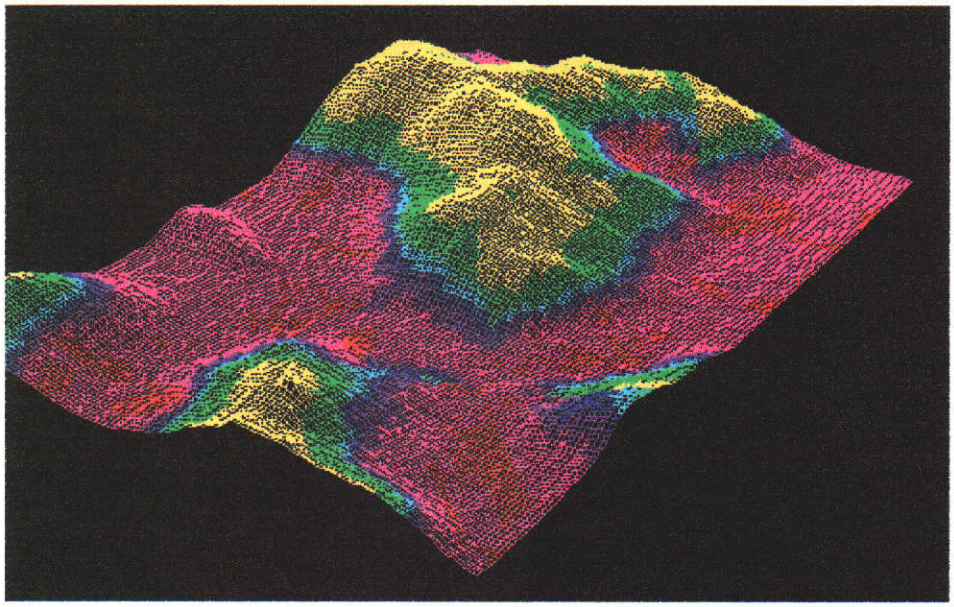


Figure 5.11: The partitioning shown in figure 5.10 overlaid on the DEM to give a three dimensional perspective. Note that classes less than 1m above the nearest drainage pixel lie in the lower parts of the landscape whereas those greater than 10m above the nearest drainage pixel tend to be hill tops or ridges.

| HeightAbove (m) | Pixel Counts | | |
|---|---|---|---|
| | Saline | NotSaline | Total |
| >10 | 0 | 5143 | 5143 |
| 5-10 | 0 | 3597 | 3597 |
| 4-5 | 0 | 861 | 861 |
| 3-4 | 3 | 857 | 860 |
| 2-3 | 9 | 1121 | 1130 |
| 1-2 | 54 | 1340 | 1394 |
| 0-1 | 8213 | 5900 | 14113 |
| < 0 | 3001 | 1262 | 4263 |
| Total | 11280 | 20081 | 31361 |

Table 5.1: Relationship between HeightAbove and Salinity.

| HeightAbove (m) | $P(Salinity \mid HeightAbove)$ | |
|---|---|---|
| | Saline | NotSaline |
| > 10 | 0.00 | 1.00 |
| 5-10 | 0.00 | 1.00 |
| 4-5 | 0.00 | 1.00 |
| 3-4 | 0.00 | 1.00 |
| 2-3 | 0.01 | 0.99 |
| 1-2 | 0.04 | 0.96 |
| 0-1 | 0.58 | 0.42 |
| < 0 | 0.70 | 0.30 |

Table 5.2: Relationship between HeightAbove and Salinity.

such the extents of this class are poorly defined. For mapping salinity, the eight classes could effectively be grouped into 2 classes: $< 1m$ and $> 1m$ above the skeletal drainage network.

A problem with the approach is that the use of flat-earth euclidean distance introduces some artifacts into the valley segments extracted. For example, the height that a pixel is above the nearest drainage network may not be the drainage network associated with the watershed that the pixel is a part of. This scenario is depicted in figure 5.12. In the figure, the thick black curve represents the cross section of (fictitious) land that is closest to $V_2$ and the thin curved line the land which is closest to $V_1$. In this scenario, the *height above* values for regions represented by the thick black line are calculated relative to $V_2$. Ideally, the *height above* calculation for the proportion of the thick black line that falls in the watershed of $V_1$ should be calculated relative to $V_1$, not $V_2$. The belief here is that the morphology of the land provides the main source of hydrological control on the (sub)surface flow of water and hence the occurance of salinity, and in this way the *height above* calculation should be performed within a given watershed. One consequence of this problem is that pixels relatively high in the watershed (associated with $V_1$) may have low (or even negative!) HeightAbove values, which would (falsely) indicate a high potential for being saline.

Another potential drawback of the technique is that it relies on the single-flow water accumulation algorithm to extract the skeletal drainage network, which would be expected to perform more poorly the flatter the terrain becomes. The performances of single-flow and multiple-flow accumulation algorithms are examined in the next section.

### Single-Flow vs Multiple-Flow Water Accumulation Algorithms

Various aspects of single-flow versus multiple-flow water-accumulation algorithms have been considered by Quinn et al. (1991) in the context of the differences in model outputs when they are applied to a single DEM. Qualitative comparisons

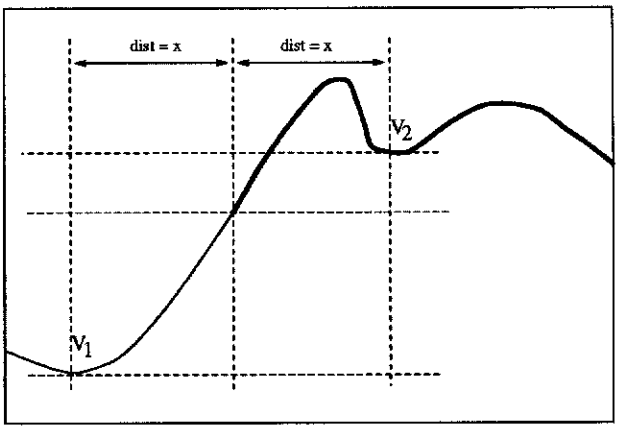



Figure 5.12: A graphical depiction of a limitation of the `HeightAbove` algorithm. This figure depicts a cross-sectional view of a ficticous landscape. In this figure, the segment of the thick black line which falls within the watershed of $V_1$ will have *height above* values calculated relative to $V_2$, whereas values relative to $V_1$ are required.

of the two models are given by Quinn et al. (1991), with the most important distinction being that as the terrain becomes flatter, the multiple-flow algorithm produces flow paths that increasingly become more dispersed whereas the single-flow algorithm attempts to force flow in one direction. In the extreme case, if water enters a flat (flood) plain with undefined creek-lines, then it would be expected that the water would disperse across the plain suggesting that in this limiting case a multiple-flow algorithm is a better model. On the other extreme, when flow-paths are well defined, for example an incised valley, then the models behave similarly.

To graphically illustrate the differences in the single-flow and multiple-flow algorithms, the images obtained for the broad-scale data were stratified into five classes (see figures 5.13 and figures 5.14 respectively): HillTops (yellow), Ridges and UpperSlopes (green), UpperValleys (cyan), MidValleys (blue) and LowerValleys (red). From the figures we observe that the single flow accumulation algorithm produces a skeletal drainage network as compared to the *fuzzy* drainage network produced by the multiple-flow algorithm.

Quinn et al. (1991) does not address how well the two models respond to poor

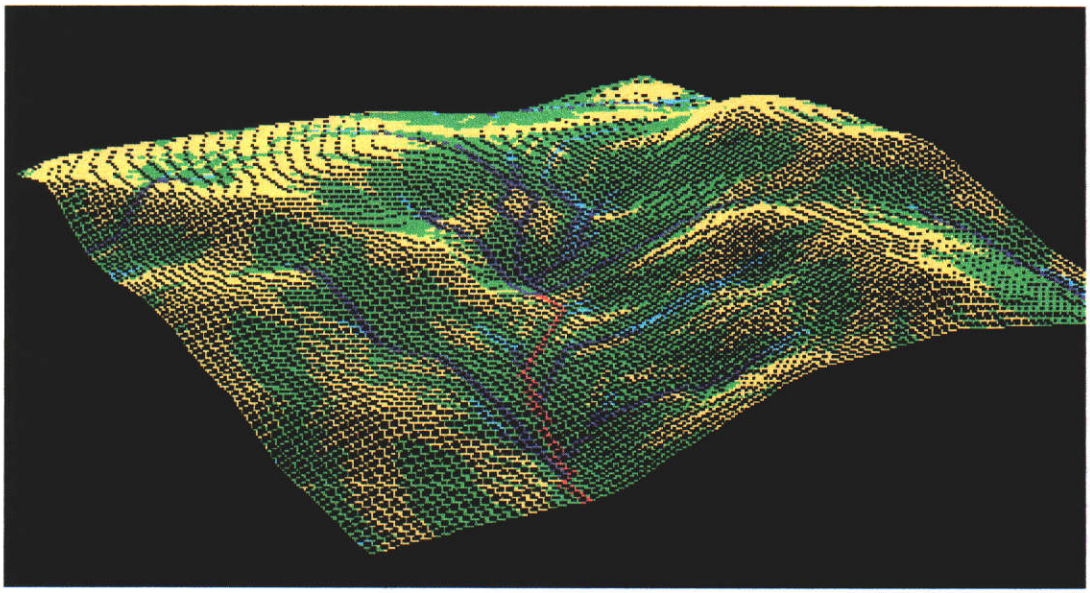


Figure 5.13: Three dimensional depiction of the results obtained from the single-flow accumulation algorithm. Here the accumulation image has been draped over the broad-scale DEM. The figure is coloured red, representing the areas of highest accumulation, through blue, cyan, green and yellow, which depicts the regions of lowest accumulation.

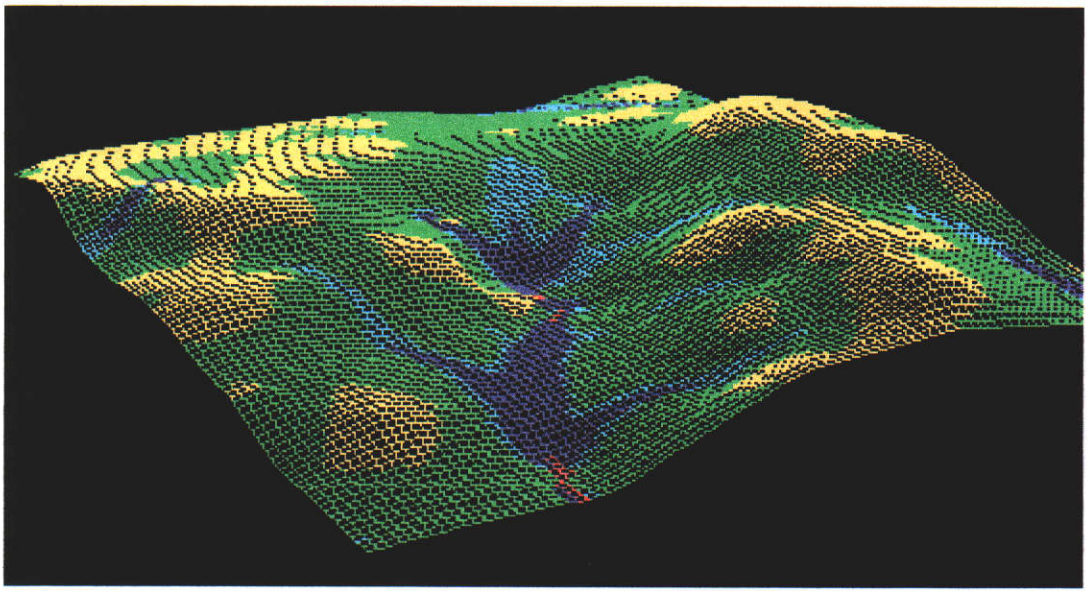


Figure 5.14: Three dimensional depiction of the results obtained from the multiple-flow accumulation algorithm. Here the accumulation image has been draped over the broad-scale DEM. The figure is coloured red, indication the areas of highest accumulation, through blue, cyan, green and yellow, which depicts the regions of lowest accumulation.

elevation definition in the DEM, as is the case here with the broad-scale DEM. This is briefly examined here.

Firstly, the sensitivity (to errors in the DEM) of the accumulation values produced by the single-flow and multiple-flow algorithm is examined.

The approach taken was:

1. Apply the single-flow and multiple-flow accumulation algorithms to the accurate DEM to produce the two accumulation images $A$ and $B$ respectively.

2. Perturb the DEM by adding random *noise* to each pixel and then re-apply the single-flow and multiple-flow algorithm to produce images $C$ and $D$ respectively. Here *noise* is introduced into the DEM by adding to each pixel, a random number in the range $[-h, +h]$. The values for $h$ used ranged from $h = 1m$, which reflects the error inherent in the accurate DEM, to $h = 5m$, a relatively large number.

3. Summarise and then compare the differences in the accumulation values generated from the original and perturbed DEM; here the differences were summarised by calculating the mean and variance of the quantities $|C - A|$ and $|D - B|$.

4. Repeat the above steps a number of times: here the above steps were performed one hundred times for each value of $h$. The results are given in table 5.3.

For the one hundred cases with $h = 1m$, the average values for the mean (variance) of $|C-A|$ and $|D-B|$ were 5.6 (360) and 5.9 (159) for the single-flow and multiple-flow algorithm respectively. For all values of $h$, the average mean error in the accumulation values resulting from the single-flow and multiple-flow algorithm are roughly within one accumulation value of each other, whereas the average variance of the results produced by the single-flow algorithm is roughly double that produced by the multiple-flow algorithm. The larger variance produced by

| | Multiple-flow | | Single-flow | |
|---|---|---|---|---|
| h (m) | Average | Variance | Average | Variance |
| 1 | 5.6 | 159 | 5.9 | 360 |
| 2 | 6.6 | 190 | 5.8 | 378 |
| 3 | 6.9 | 196 | 5.8 | 384 |
| 4 | 7.0 | 199 | 6.2 | 387 |
| 5 | 6.6 | 191 | 6.2 | 378 |

Table 5.3: Comparison of the sensitivity of the single-flow and multiple-flow algorithm to errors in the DEM.

the single-flow algorithm demonstrates that this algorithm is more sensitive to errors in the DEM as compared to the multiple-flow algorithm.

To examine whether this effect is observed when comparing the accumulation values obtained from the accurate DEM and the broad-scale DEM, the following was performed:

1. Apply the single-flow accumulation algorithm to the region where both the accurate and broad-scale DEM are defined to form the respective images *SingleFlow1* and *SingleFlow2*. Do likewise for the multiple-flow algorithm to form the respective images *MultipleFlow1* and *MultipleFlow2*.

2. Form the two *error* images *SingleFlowError* = |*SingleFlow1* - *SingleFlow2*| and *MultipleFlowError* = |*MultipleFlow1* - *MultipleFlow2*| representing the absolute errors between the algorithm applied to the accurate and inaccurate DEM respectively.

From the images, the average error (and variance of the error) for the region, shown in figure 5.13, were calculated and are given in table 5.4. We observe that the average errors obtained from the single-flow and multiple-flow algorithms are roughly equal and the variance in accumulation values obtained from the single-flow algorithm is significantly higher than that obtained from the multiple-flow algorithm. This is consistent with the results obtained from perturbing the DEM

| Algorithm | Statistics (units = accumulation) | | |
|---|---|---|---|
| | Average | Variance | Pixels |
| SingleFlowError | 28 | 32160 | 68628 |
| MultipleFlowError | 26 | 9153 | 68628 |

Table 5.4: Comparison of the single-flow and multiple-flow algorithm to poor elevation definition in the broad-scale DEM.

and further demonstrates that the multiple-flow accumulation algorithm is less sensitive to errors in the DEM.

The relatively high errors were largely the result of both methods performing poorly in flatter regions, where small undulations in the DEMs result in significant changes in the flow paths (problems in these areas are specifically addressed in section 5.4). The relatively large variances result from parts of the main drainage system being located in different positions in the two DEMs. This is in part due to lack of broad-scale elevation data in these regions for the creation of the broad-scale DEM and, consequently, poor valley minima definition.

From the above experiment, it appears advantageous to use the multiple-flow algorithm for extraction of local drainage networks, as the results are less likely to be affected by poor definition in the DEM. For both algorithms, drainage definition in the broad, flat valleys was poor. These areas are the topic of the next section.

## 5.4 Flat-Area Classification

### 5.4.1 Existing Approach

The flat area classifier descibed by Chorowicz et al. (1989) (and discussed in section 2.5.4) was implemented and applied to the DEM study area. As defined by Chorowicz et al. (1989), a broad-valley label is given to pixels having the configuration [$--$ $==$ $++$]. This definition may be extended (Chorowicz et al., 1989) to include valleys which are wider than two pixels. The implementation

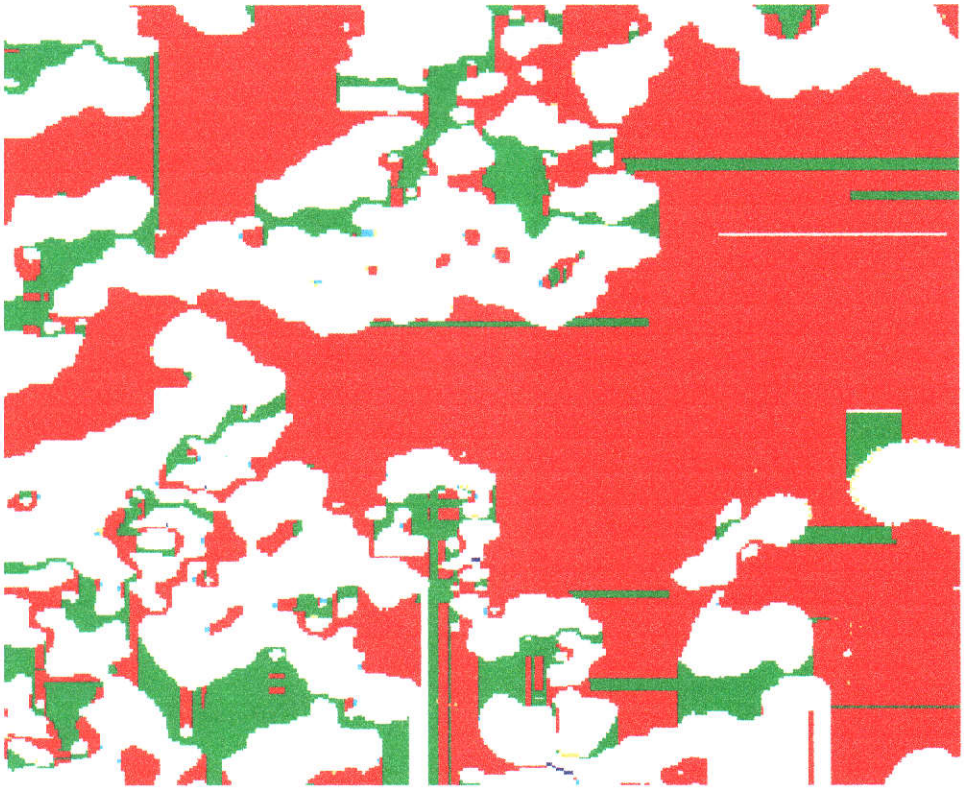


Figure 5.15: Flat area classification of the broad-scale DEM produced using the algorithm described by Chorowicz et al. (1989). Flat hill tops and ridges, flat valleys, flat inflextions and non-flat regions are coloured green, red, blue and white respectively.

used here considers a variable length flat region $[-- = \ldots = ++]$ as a broad valley. In this way, broad valleys of multiple scale can be extracted.

The algorithm was used to classify the broad-scale DEM into the classes: flat hilltop/ridge, flat valley, flat inflexion and not flat, which are coloured green, red, blue and white respectively in the example given in figure 5.15. For visual comparison with earlier methods, the classification is shown draped over the DEM in figure 5.16.

From figures 5.15 and 5.16, the spatial weakness of this approach is evident; pixels within flat regions are inconsistently labelled in relation to their neighbours, for example in figure 5.15 straight green stripes representing hill tops/ridges have large neighbouring areas labelled as flat valleys. The reverse situation is also true.

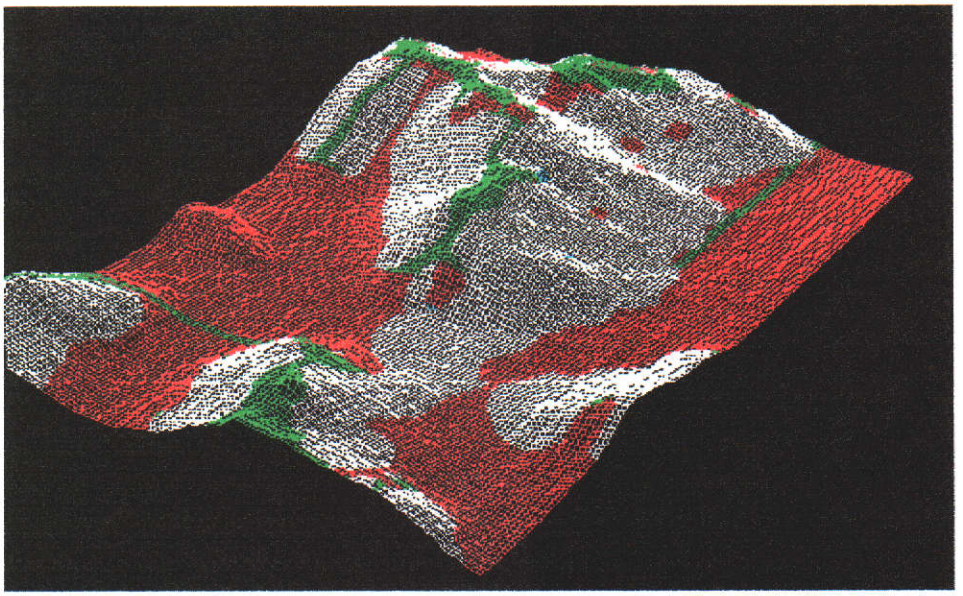


Figure 5.16: Three dimensional depiction of the flat area classification of the broad-scale DEM produced using the algorithm described by (Chorowicz et al. 1989). The colours are the same as for figure 5.15.

This problem arises because errors in the DEM in the direction of the scan can effect the labelling of many pixels in the scan. For example, if a scan produced the profile string [—=====++++], then all '=' symbols would be labelled as belonging to a broad valley. If however the scan produced the profile string [—+=====-++++] then all '=' symbols would be labelled as belonging to a flat hill top: this effect can be caused by factors including: errors in the elevation data and small overshoots and other errors introduced in the interpolation process.

### 5.4.2 New Approach

This limitation suggested that a better flat-area classifier could be developed if the properties of the flat areas were collected on a feature scale, largely invariant of feature size. Using this idea, a broad valley is characterised as a *flat area mostly bounded by pixels higher than it*, whereas at the other extreme a feature such as a flat hilltop will be characterised as a *flat area bounded by regions lower that it*. This characterisation was used to implement a new flat-area classifier called

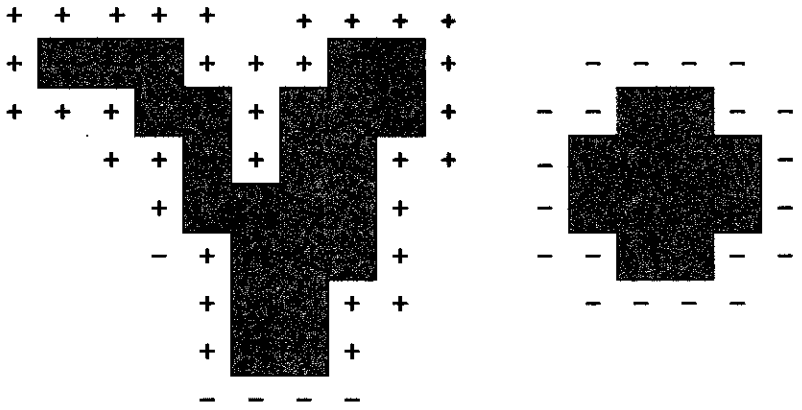


Figure 5.17: New feature characterisations: (left) broad valley and (right) flat hilltop. The grey region represents the feature of interest and the positive and negative symbols indicate that a bounding pixel is higher than or lower than the feature respectively.

INFLOWS in the following.

The algorithm calculates for each flat area the percentage of boundary pixels that flow into the region, where the inflow value ranges from one hundred, indicating all pixels flow into the region to zero, indicating no bounding pixels flow into the region. An artificial example illustrating a broad valley segment and a flat hilltop is given in figure 5.17.

This variable may then be used as a continous variable or to classify the flat areas into classes such as hilltops, inflections and broad valleys, with the latter class being of most interest.

The algorithm (see Appendix A for algorithm code) developed performs the following operations:

1. The DEM is read and a slope direction map is constructed. For each pixel, the slopes to its eight neighbours are computed and then the pixel is set to the sign of the slope bearing the greatest magnitude. For this algorithm, only the sign of the slope is relevant, thus a slope direction map composed of three symbols $\{+, -, =\}$ is generated. Flat areas are defined by a slope threshold; pixels with slope less than the threshold are given a '=' symbol.

2. The slope direction map is scanned until a '=' symbol is found. This pixel

acts as a seed pixel to a modified flood-filling algorithm based on (Kirk, 1992), which will be referred to as `FloodFill` in the following. A flood-fill is performed and counts $N(+)$ and $N(-)$ of the respective symbols '+' and '−' found at the boundary of the region during the fill are made.

3. The ratio $R = N(+)/(N(+) + N(-))$ representing the proportion of pixels flowing into the region is calculated, and then the flat region is refilled with this value. This process is repeated for each flat region found. On completion, flat regions may be classified according to their $R$ value.

The critical aspect of applying the algorithm is the choice of slope threshold, as this determines how much of a given area is considered flat. The flat-area classifier was applied to the DEM using various slope thresholds and the resulting partitioning examined. The effects observed when varying the slope can be illustrated by examining the results obtained from the thresholds 0.01, 0.02, and 0.03, shown in figures 5.18, 5.19 and 5.20 respectively. The figures have the same colour table and are coloured red, indicating a high $inflow$, through orange, yellow and green which indicates a low $inflow$.

From these figures, it is observed that a higher slope threshold produces larger flat areas. More importantly, some of the flat areas join to become one area as the slope threshold is increased, which may significantly change the $inflow$ characteristics of the features extracted. Too low a slope threshold (for instance 0.01) produced flat areas, the shapes of which had artifacts of the interpolation process in them, for example square boundaries that are the result of the GRASS routine generating the DEM in tiles. Too large a slope threshold identified a large proportion of land as being flat, and changed the inflow characteristics of the regions.

To examine which threshold is better, information obtained from the $inflow$ algorithm was compared to information on salinity. In particular, the flat-valley segments were extracted from the $inflow$ image (by specifying that valley segments have $R > 0.6$) and a new image representing areas that are flat-valleys

| Flat-valley image | Salinity ground truth | |
|---|---|---|
| | Salt | NotSalt |
| | Slope = 0.01 | |
| Flat-valley | 9570 | 3366 |
| Not flat-valley | 1433 | 13713 |
| | Slope = 0.02 | |
| Flat-valley | 10607 | 3914 |
| Not flat-valley | 396 | 13165 |
| | Slope = 0.03 | |
| Flat-valley | 10735 | 4358 |
| Not flat-valley | 268 | 12721 |
| | Slope = 0.05 | |
| Flat-valley | 10838 | 4993 |
| Not flat-valley | 165 | 12086 |

Table 5.5: Coincidence table of salinity information and the flat-valley segments generated using different slope thresholds. The numbers are pixel counts.

and are not flat-valleys generated. This new image was compared with salinity information and the results are given in table 5.5. Assuming that the coincidence tables should only have diagonal entries (this corresponds to assuming that only flat-valleys are saline; a reasonable assumption for this area), then flat-valleys are identified with an accuracy of 82.9%, 84.7%, 83.5% and 81.6% for the slope thresholds of 0.01, 0.02, 0.03 and 0.05 respectively. On this basis a slope threshold of 0.02 is chosen for subsequent use of this algorithm.

Comparing figure 5.19 with figure 5.15, it is obvious that this new approach is spatially superior to that of figure 5.15, having no inconsistencies in the resulting interpretations of each flat region.

### Combining Accumulation Classes with Flat Areas

Next, a partitioning of the landscape was generated by patching together the broad-valley segments extracted using the INFLOWS algorithm with the classes derived from the multiple-outflow water accumulation algorithm. An example of

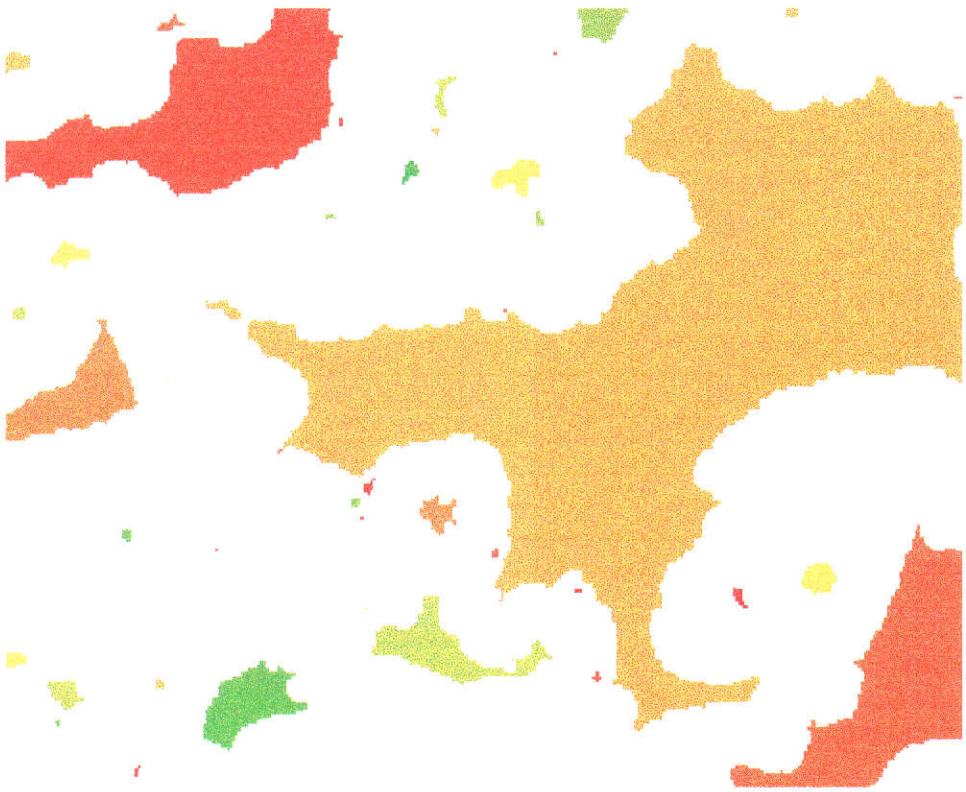


Figure 5.18: Results of the INFLOWS algorithm. Slope = 0.01. The regions are coloured green for $R < 0.1$, through to yellow, orange and then red for $R > 0.9$.

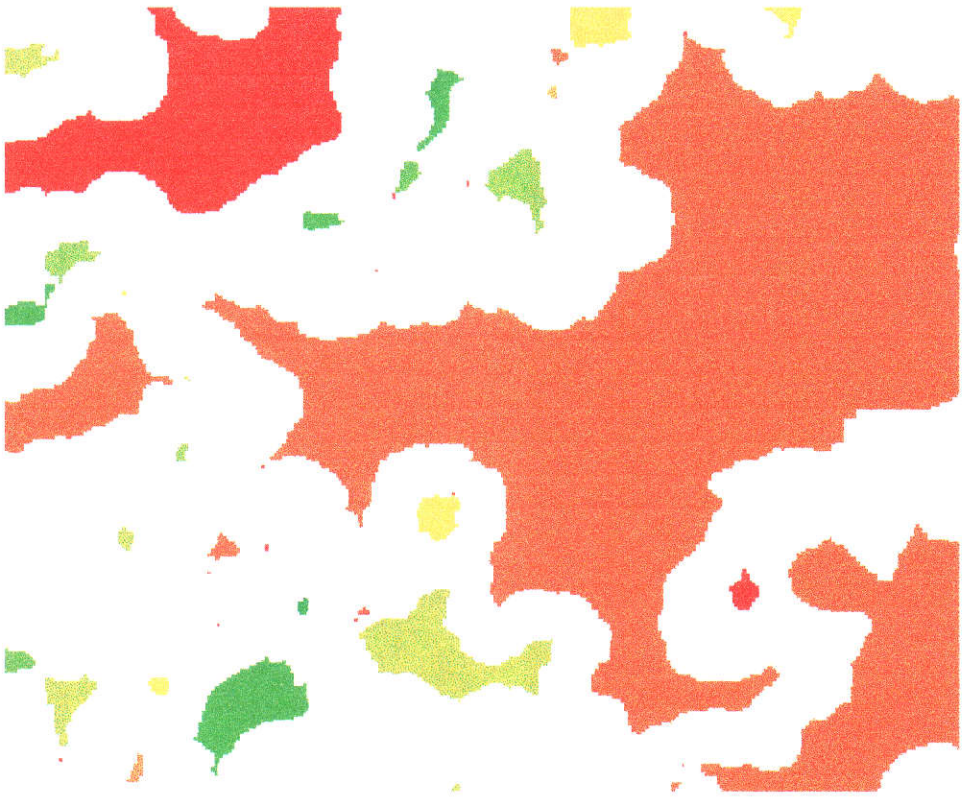


Figure 5.19: Results of the INFLOWS algorithm. Slope = 0.02. The regions are coloured green for $R < 0.1$, through to yellow, orange and then red for $R > 0.9$.

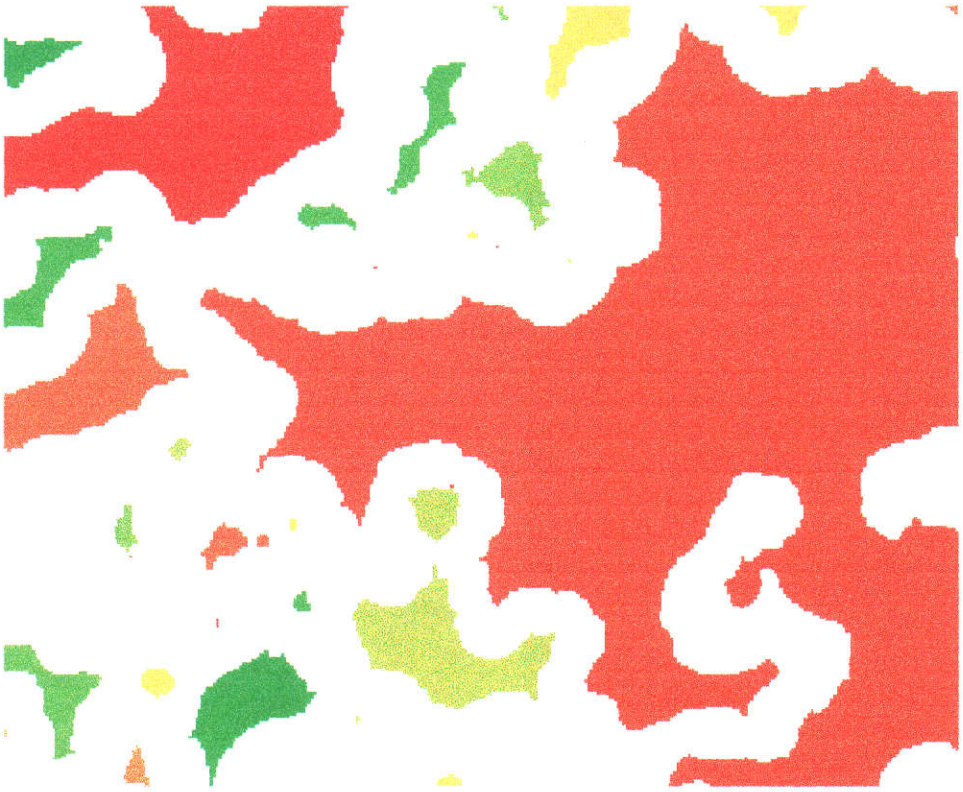


Figure 5.20: Results of the INFLOWS algorithm. Slope = 0.03. The regions are coloured green for $R < 0.1$, through to yellow, orange and then red for $R > 0.9$.

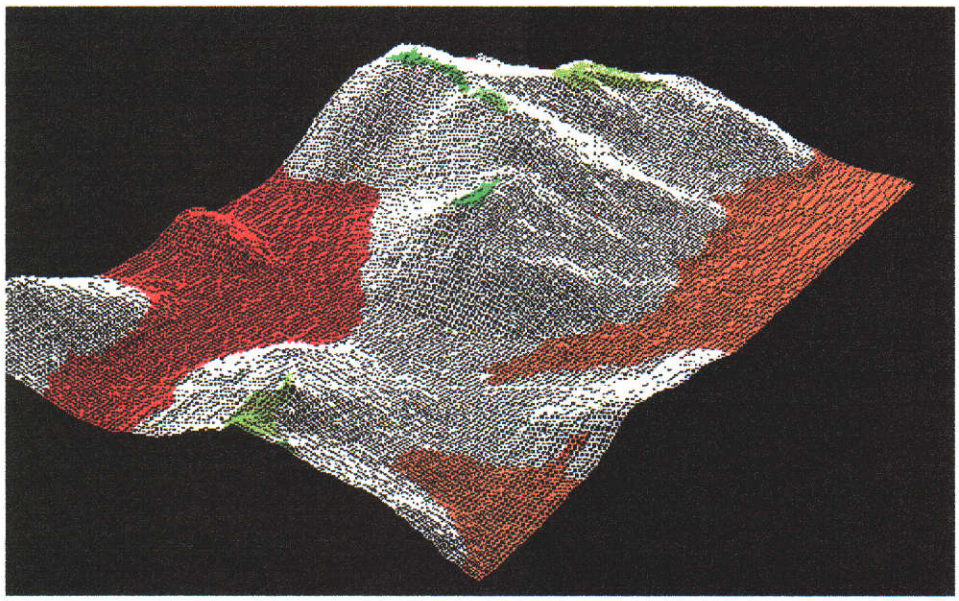


Figure 5.21: A three dimensional perspective of the results of the INFLOWS algorithm (slope = 0.02). The INFLOWS results are draped over the DEM.

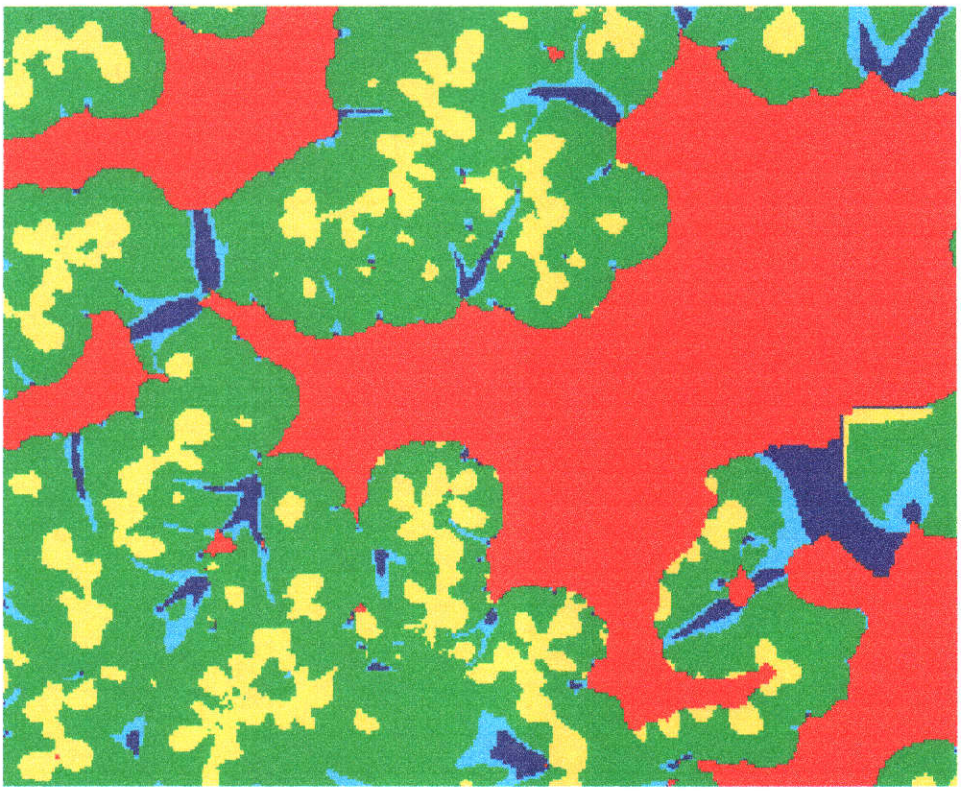


Figure 5.22: Results of the combined partitioning. Here the colours yellow, green, cyan, blue and red represent the classes HillTops, Ridges / UpperSlopes, Upper-Valleys, Lower Valleys and BroadValleys respectively.

the result is depicted in figure 5.22. A limitation of this approach is that segments of broad valley that are not identified as being flat are also not identified as being valleys by the accumulation algorithm. This approach is remedied in the next section, by considering flat areas as being regions that also exhibit flow.

The partitioning was then related to data on salinity to form the cross tabulation of salinity versus landform given in table 5.6 from which the relationships in table 5.7 were derived. We observe that the partitioning again provides relatively strong evidence for where salinity is not; $P(NotSaline|HillTops) = 0.99$, $P(NotSaline|Ridges/UpperSlopes) = 0.97$, $P(NotSaline|UpperValleys) = 0.89$, $P(NotSaline|LowerValleys) = 0.7$ and relatively strong information where salinity is; $P(Saline|BroadValleys = 0.73$. The probabilistic assessments are useful when considering reasoning with Bayesian networks. For example, although areas

| Landform | Pixel Counts | | |
|---|---|---|---|
| | Saline | NotSaline | Total |
| Hilltops | 37 | 3268 | 3305 |
| Ridges/UpperSlopes | 318 | 11761 | 12079 |
| UpperValleys | 79 | 668 | 747 |
| LowerValleys | 152 | 358 | 510 |
| BroadValleys | 10712 | 3968 | 14680 |
| Total | 11298 | 20023 | 31321 |

Table 5.6: Cross tabulation of salinity and the landfrom classes defined using the approach presented in section 5.4.2.

| Landform | $P(Saline\|Landform)$ | |
|---|---|---|
| | Saline | NotSaline |
| Hilltops | 0.01 | 0.99 |
| Ridges/UpperSlopes | 0.03 | 0.97 |
| UpperValleys | 0.11 | 0.89 |
| LowerValleys | 0.30 | 0.70 |
| BroadValleys | 0.73 | 0.27 |

Table 5.7: The relationship between salinity and landform as derived from the counts given in table 5.6.

identified as UpperValleys tend not to be saline, $P(Saline|UpperValleys) = 0.11$, they are approximately ten times more likely to be saline that those areas identified as Hilltops. Bayesian models accomodate this information in the reasoning process.

## 5.5 New Approach — Combined Accumulation Algorithm

Based on the observation that segments of the valley system were not identified by the previous algorithm, the following multiple-flow accumulation algorithm has been devised, which incorporates multiple-flow regions into the per-pixel multiple-

outflow model.

The methodology behind this approach is to use the per-pixel multiple-outflow model where per-pixel outflows are adequately defined and to use the flow characteristics of regions where per-pixel outflows are poorly (or not) defined.

### Considering Flow in Flat Regions

The flat features of most interest in the hydrological sense are the broad valley systems that are common in W.A. These features are extracted using the INFLOWS algorithm introducted in the previous section.

On a per-pixel basis, flow characteristics in the flat-broad valley segments are either not defined or very poorly defined. For this reason, these regions are considered here to be entities which possess broad-scale flow characteristics, with the assumption that the flow characteristics within the regions are unobservable from the DEM. This is a reasonable assumption under the following scenarios:

1. A valley segment may truly be flat, in which case any surface flow takes the form of *sheet* flow.

2. A valley segment may not really be flat, but is flat due to the inaccuracies in the DEM. In this case there may be one distinct flow path within the region but there is no information as to where this flow line is. We therefore acknowledge that any pixel in this region is equally likely to be the flow path.

Flow is considered to enter a flat region from (non-flat) boundary pixels that are *higher* than it, and exit the region at (non-flat) boundary pixels *lower* than it. Here, *higher* and *lower* are local relative measures as used in section 5.4.2. Thus individual pixels which are higher than the region pass their flow to the flat area and then the flat area distributes this flow to individual pixels lower than it. This is depicted in figure 5.23, where arrows indicate the direction of flow.

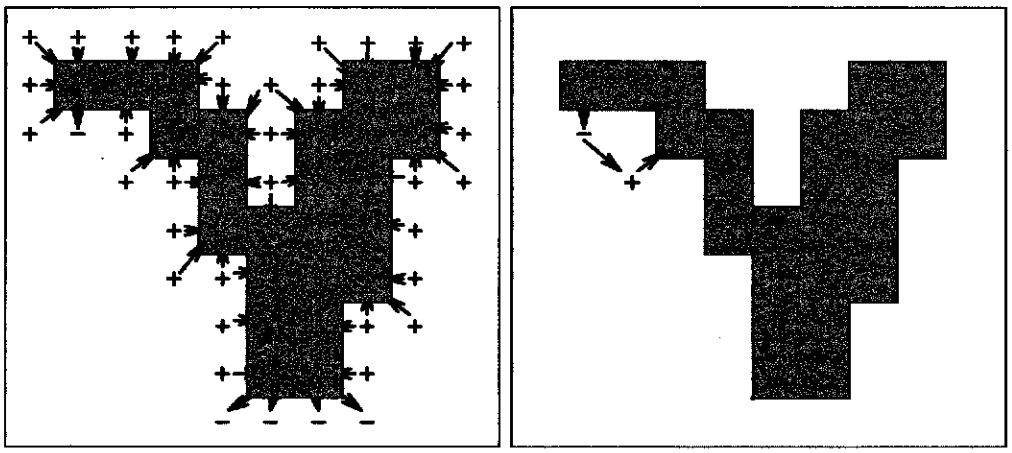


Figure 5.23: A flat region, coloured grey, with a potential flow feed-back problem. In the diagrams, bounding pixels which are higher that or lower than the nearest flat feature pixel are labelled with the symbols '+' and '-' respectively. The flow characteristics of this region (left) may result in flow feed-back path shown by the arrowed path (right).

Having identified the broad-valley segments, the improved water accumulation model proceeds as follows:

1. Initialise each cell in the image to have one unit of rainfall.

2. For all pixels that are not part of a broad-valley region, distribute its accumulated water to its immediate downhill neighbours.

3. For each broad-valley region, calculate the aggregated flow into the region and distribute this flow amongst all out flow points of the region.

4. Iterate steps (2) and (3) until the model has drained.

A problem encountered with initial implementations of this algorithm was that once *water* had left a flat feature, it often flowed (by some per-pixel path) back onto the feature, as depicted in figure 5.23. This resulted in *water* circulating and the model not draining. Also, because of this, the *upslope area* estimates were not correct.

For this reason an extra step was added to the algorithm:

1.5 For each flat region, check all outflow points to see if the outflow point generates an inconsistent flow path, if it does then mark the point as not being an outflow point. In this way the inconsistency is not removed, but flow feedback is avoided.

Flow feed-back paths may be checked by performing a breadth-first search from each outflow pixel, where the search is constrained to consider only neighbouring pixels which are lower than a pixel currently in question. If the search identifies a pixel which is labelled as belonging to the flat feature in question, then the outflow point generates a flow feed-back path.

Comments:

- The aggregated inflow calculation in (3) is achieved by summing all units of flow into the region. A modified flood-fill algorithm (Kirk, 1992) was used to scan the features.

- Upon completion, each pixel in the broad-valley region has a value equal to the amount of water that flowed to the entire region. Each pixel not in such a region has a label equal to the amount of water that flowed to it.

- Flat regions with no outflows do not drain. In practice, lakes represent such a circumstance.

- The choice of slope threshold must be made with some care, as too large a threshold will identify too much of the landscape as flat, thus creating drainage regions that potentially flow onto themselves.

- Flow feed-back is identified by a local search strategy: *for each outflow pixel, check every potential flow path for the possibility of flow feed-back.*

The pseudo code for the above algorithm, which will be called ALLFLOW3, is given in Appendix A.

The ALLFLOW3 algorithm was applied to the data to produce an accumulation image, which was then stratified in the same manner as in section 5.4.2 to

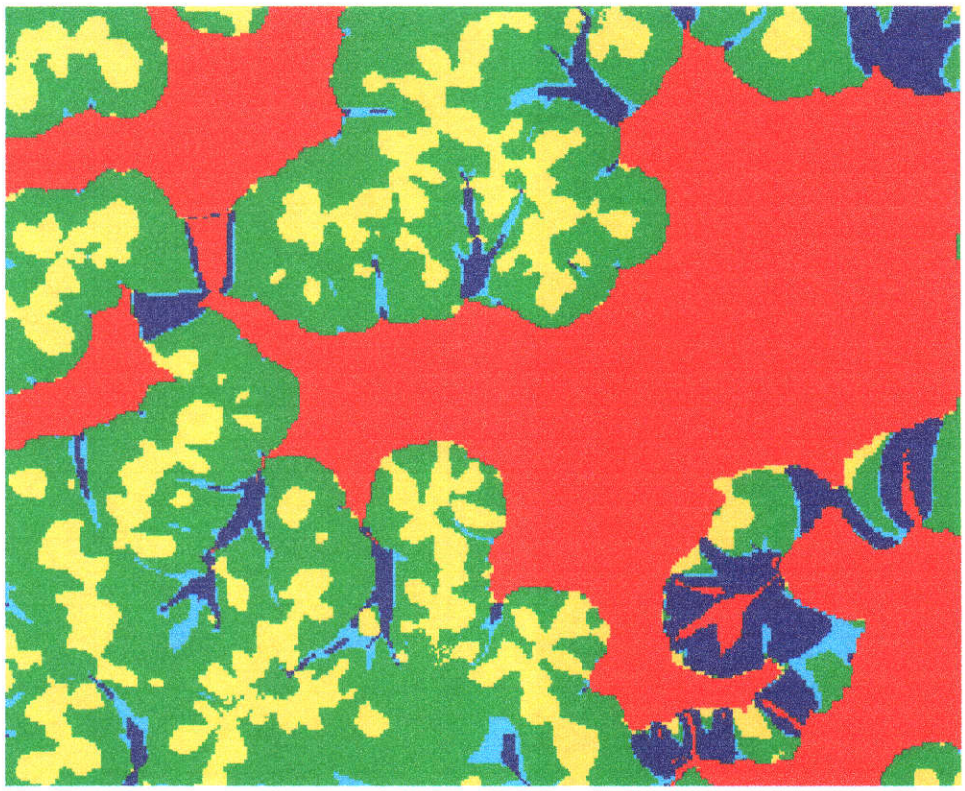


Figure 5.24: Example of ALLFLOW3 Partitioning.

obtain landform classes. An example of the partitioning is given in figure 5.24 and 5.25, and the partitioning for the MKG area in figure 5.26.

To determine the performance of the landform partitioning obtained from ALLFLOW3, the partitioning was related to data on salinity to form the cross tabulation of salinity versus landform, given in table 5.8, from which the relationships in table 5.9 were derived.

From table 5.9 we observe that the partitioning provides relatively strong evidence for where salinity is not; $P(NotSaline|HillTops) = 0.99$, $P(NotSaline|Ridges/UpperSlopes) = 0.98$, $P(NotSaline|UpperValleys) = 0.91$, $P(NotSaline|LowerValleys) = 0.75$ and relatively strong information where salinity is; $P(Saline|BroadValleys = 0.73$.

The relative performance of the ALLFLOW3 algorithm and the simpler approach described in section 5.4.2 can be made by comparing tables 5.6 and 5.7

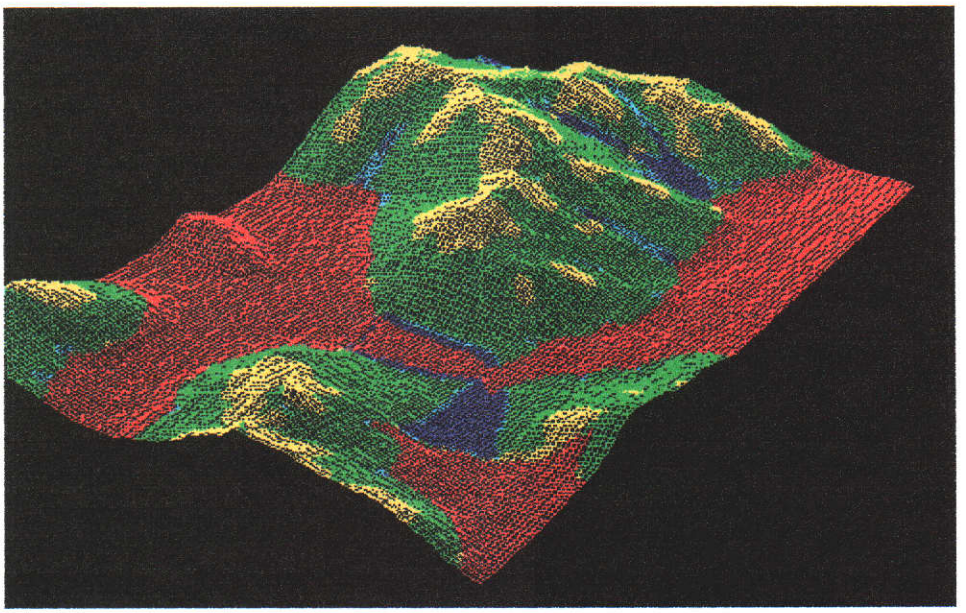


Figure 5.25: Example of ALLFLOW3 Partitioning: three dimensional perspective view.

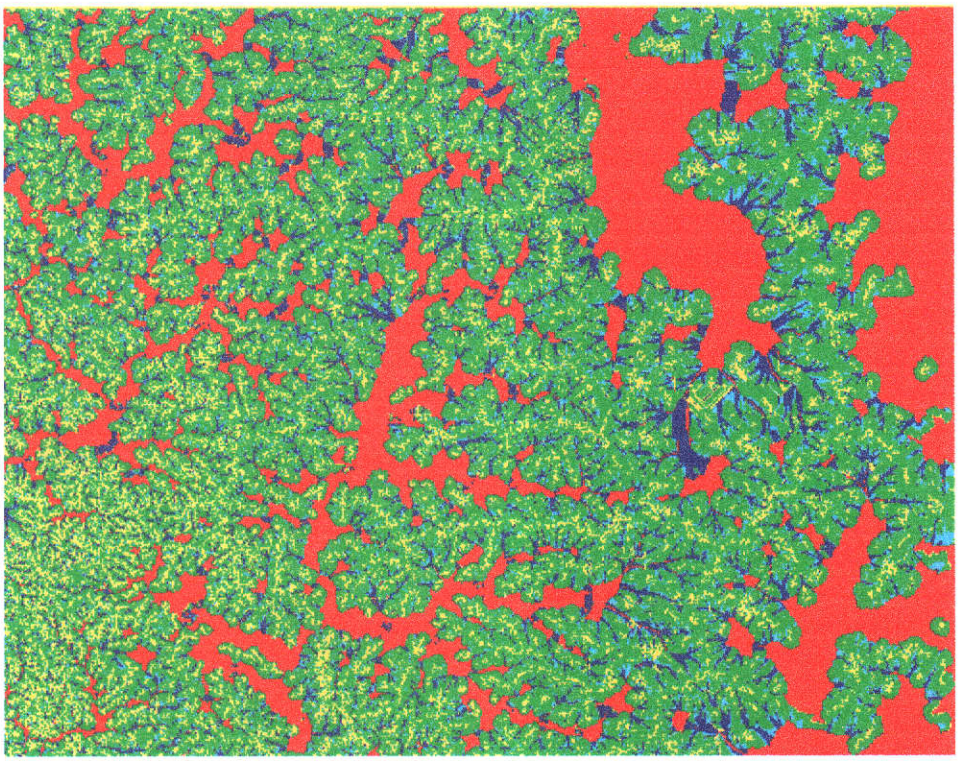


Figure 5.26: Partitioning of the MKG Study Area.

| Landform | Pixel Counts | | |
|---|---|---|---|
| | Saline | NotSaline | Total |
| Hilltops | 56 | 4569 | 4625 |
| Ridges/UpperSlopes | 240 | 10046 | 10286 |
| UpperValleys | 60 | 643 | 703 |
| LowerValleys | 233 | 706 | 939 |
| BroadValleys | 10709 | 4059 | 14768 |
| Total | 11298 | 20023 | 31321 |

Table 5.8: Cross tabulation of salinity and the landfrom classes defined using the ALLFLOW3 algorithm.

| Landform | $P(Saline\|Landform)$ | |
|---|---|---|
| | Saline | NotSaline |
| Hilltops | 0.01 | 0.99 |
| Ridges/UpperSlopes | 0.02 | 0.98 |
| UpperValleys | 0.09 | 0.91 |
| LowerValleys | 0.25 | 0.75 |
| BroadValleys | 0.73 | 0.27 |

Table 5.9: The relationship between salinity and landform as derived from the counts given in table 5.8.

with tables 5.8 and 5.9 respectively. Comparing table 5.7 and table 5.9 we observe that little difference in the relationship between landform and salinity results. However, from tables 5.6 and 5.8 we observe that the LowerValley class has approximately twice as many pixels in the ALLFLOW3 results than the method described in section 5.4.2. This is not surprising given that the idea behind the ALLFLOW3 algorithm was to generate those segments of the flow path that the algorithm given in section 5.4.2 missed.

## 5.6 Summary

This chapter has considered various issues relating to deriving features from a digital elevation model. Broad-scale information on elevation is available from

the Department of Land Administration in the form of elevation sites collected at regular intervals in elevation. These *contour* data were originally derived from photo-interpretation. Depending upon the region considered, data may be collected at 5m, 10m or 20m increments in elevation.

Experiments conducted in this chapter were performed on the Moora - Kalannie /Goodlands study region. For most of the region, 10m contour data were available; however only 20m contour data were available in the eastern most extent of the region. While 10m elevation data could be considered quite good for some terrains, the flat nature of the terrain typical of the Western Australian wheatbelt means that elevation samples may be laterally separated by hundreds of metres.

The methods were developed in a region around Bindi-Bindi because *accurate* elevation data were available for this region.

Firstly, to gain some understanding of errors in the broad-scale elevation data, the data were compared with accurate elevation data. The method used for comparison relied on comparing the DEMs derived from each of the data sets. From this comparison, the largest errors were observed at the local extrema of the resulting surfaces, that is on hill tops and valley floors. This was not surprising, considering that the broad-scale data were collected at fixed contour increments, and these two regions are prone to having missing data.

While errors regarding the elevation of hilltops are not expected to be of concern for the applications considered in this thesis, the errors associated with the identification of valley floors is of concern, as these features have a tendency to go saline. For broad-valley floors, the errors have the following implications:

1. Broad-valley floors are overestimated.

2. The break of slope occurring at the location where a hillside slope is adjacent to a broad-valley system (also referred to as the *foothill*), may be displaced by hundreds of metres, with 500m displacements being possible. This has important ramifications for the detection of small scalded areas of land which

occur in these regions. Further, arguments were given for the impractibility of accurately identifying breaks of slope in general.

Having considered some aspects of the errors in the broad-scale DEM, various approaches for extracting features such as hill tops and valley floors were considered. Simple, widely available (published) schemes were considered first.

Classifying the terrain using the MLC and the DEM curvature estimates as data produced no useable results. The lack of curvature estimates for the broad-valley was also of concern[3].

A single-flow water accumulation algorithm was then applied to the data, and valley systems and ridge lines extracted. Although the valley systems were positioned in the true valley system, their skeletal representation was a poor description for the broad, flat characteristics of the true drainage system. Also, with this representation, ridge lines extended into the broad valley system.

Given that the above method identified the general locality of the valley system, it seemed plausible to perform some form of constrained region-growing to extract a better representation of the *true* broad-valley drainage system from the skeletal drainage system. To achieve this, an algorithm, called `HeightAbove`, was developed and applied to the DEM. While the algorithm initially appeared promising, it produced some unwanted artifacts including labelling pixels relatively high in a catchment with low HeightAbove (streamline) values. It also was reliant on the single-flow water accumulation algorithm, which performed poorly when applied to even flatter terrain.

A multiple-flow water accumulation algorithm was then applied, producing a *fuzzy* drainage network. This latter algorithm was also shown to be less sensitive to errors in the DEM than the single-flow water accumulation algorithm. However, a problem with defining flow in broad-flat valley regions still existed, and needed to be addressed.

[3]The author acknowledges that curvature estimates could have been obtained if a large neighbourhood was considered; however, the vastly differing spatial extents of the features would still be a problem in choosing a suitable window size to achieve this.

The first step in solving this problem considered methods for extracting broad-valley segments. Firstly, the algorithm proposed by Chorowicz et al. (1989), which was later used by the author to extract broad-valley segments (Chorowicz et al., 1992), was implemented and applied. Although this algorithm was partially successful, it produced many inconsistently labelled regions. These resulted because the algorithm identifies different landform profiles in its four different scan directions.

A new approach to extracting flat features was derived. It was based on using a modified *flood-fill* algorithm to scan the profile characteristics of flat features. In particular, flat features were labelled by the ratio of the percentage of adjacent pixels which were higher than it. In this way flat hill-tops, with a relatively low ratio, can be separated from flat valleys which have a relatively high ratio. This algorithm was called `INFLOWS` and produced consistantly labelled regions.

Broad-valley segments were extracted using the *inflow* variable. For regions not considered a broad valley, the multiple-flow water accumulation algorithm was stratified to extract features such as hilltops and non-flat valley segments. These two methods used together formed a landform partitioning which was shown to be related to salinity.

One major problem with the partitioning was noted: some segments of the valley system were not identified. These segments were not considered flat, and therefore not extracted by the `INFLOWS` algorithm, but were not identified by the accumulation algorithm as they were *downstream* of a broad-valley segment, causing the water accumulation algorithm to perform poorly in locating the missing segments. A combined water accumulation algorithm, called `ALLFLOW3`, was devised to address this problem. The algorithm considers broad-valley regions as flow objects, much as a single pixel is a flow object in the normal water accumulation algorithm. With this algorithm, the *water* flows unimpeded across the DEM surface in much the same way as would be expected in flood-like conditions. With this method, the final accumulation value can also be interpreted as the area of

land upslope of a given pixel.

# Chapter 6

# Experiments in Combining Data — Moora/Kalannie-Goodlands

## 6.1 Introduction

Remote sensing (RS) and geographical information system (GIS) technologies have been shown to form the basis for cost-effective, broad-scale assessments of many environmental monitoring problems, including landuse and land condition assessments (Wilkinson et al., 1992; Kontoes et al., 1993; Wharton, 1987) forestry management (Skidmore, 1989; Behn and Campbell, 1992) and mineral exploration (Kruse et al., 1993).

From a remote sensing perspective, the interest generally is in quantifying the information that a specific sensor can provide, and then using the data to provide information over a broad area. Techniques such as those described by Kiiveri and Campbell (1992) that are specific to one form of data are typically used, and the results are expressed as maps of labels.

As is evident in the expanding literature on mapping applications, one specific data set may not be sufficient for a given application; multiple data sets may have to be considered. A problem that naturally arises is under what scheme should the data be combined.

Models are investigated here which enable integration of multiple sources of information with the purpose of classifying pixels or making probabilistic assessments about the label (class) of pixels in an image.

Because some variables in the models are not directly observed (that is, the the state of some of the variables is inferred from other variables), an expectation-maximisation (EM) algorithm is used to estimate the parameters of the models. Some comparisons with the more obvious approach (equation 4.4) are made.

To test the validity of the approach, simple models that combine single-date remotely sensed data and ground data are applied as described in section 6.3. In particular, soil information was combined with remotely sensed Landsat TM data. Firstly, the model considered by Strahler (1980) was applied and compared with the MLC. The model of Strahler (1980) was applied in a new way: by making use of the EM algorithm for parameter estimation. Next, the model of Strahler (1980) was extended by incorporating estimates of error rates of the spectral component.

Although the models considered in section 6.3 are relatively simple, they provide insights into the following aspects of combining data:

1. The use of Bayesian approaches to combining data.

2. The benefits, if any, and the effects on the resulting land cover interpretation obtained by combining remotely sensed data with other data.

3. The benefits, if any, of incorporating data source error estimates in the models.

4. The benefits, if any, of estimating the model parameters using the EM algorithm as opposed to more obvious techniques.

The models considered in section 6.3 are deficient in a number of ways, including:

1. Only two sources of data are combined.

2. Temporal rules are not considered.

3. Spatial rules are not considered.

To examine the effects of the above on the resulting land cover interpretation, models incorporating temporal and spatial rules are considered in section 6.4. In these models, ground data are explicitly ignored so that the effects of the temporal rules can be examined. Spatial rules are then incorporated into the model. The interpretations resulting from the temporal/temporal-spatial models are then compared with each other and also with the models presented in section 6.3.

Based on an understanding of the likely effects of combining ground data with remotely sensed data, and of combining temporal remotely sensed data in *naive* ways, in section 6.5, a more complex model for combining multiple sources of data is devised for the purpose of salinity mapping. In particular, it is demonstrated that the domain knowledge may be used to form a model specific to an application. This application also examines the feasibility of using the generalised probability propagation scheme in a more complex model. The benefits of using this approach include: many different and potentially complex models may be devised (graphically) without recoding the models explicitly; multiple sources of data may be combined simultaneously; missing data are accommodated; and if mapping accuracy is acceptable, then the Bayesian network forms the basis of a probabilistic expert system which may be used to answer what-if scenarios.

Because of the availability of data, the following experiments were conducted in the Xantippe catchment situated roughly in the centre of the MKG study area.

## 6.2 The Xantippe Catchment

The main landuse in the area is the growing of cereal grains (for example wheat and lupins) and pastures for grazing sheep. The area has fairly low average annual rainfall (330mm) and a proportion of relatively unproductive sandy soils, resulting in poor crop and pasture growth for some soil types.

The clearing of land for agricultural production has resulted in rising saline

ground water and salinisation of previously productive land. Dead trees, highly salt-tolerant species such as samphire, or bare salt-scalds are typically associated with severely salt-affected land. Moderately salt-tolerant species such as barley grass may be associated with marginally salt-affected land. These landcovers may be mapped using Landsat TM data to give broad-scale estimates of the extent of salinisation (Wallace and Wheaton, 1990; Wheaton et al., 1992, 1994). Typically, maximum-likelihood classification (Campbell and Wallace, 1989) is used.

Another indicator of the onset of salinity is poor crop or pasture growth. These landcovers may also be mapped using Landsat TM data (Furby et al., 1995), although ways of distinguishing poor growth due to salinity and other effects such as wind and water erosion, poor soils and farm management practices such as intensive grazing need to be devised. Some applications which address these issues are considered here and have been reported elsewhere by Caccetta et al. (1995a).

In the following, all spectral class estimates have been previously derived, and are described by Furby et al. (1995).

## 6.3 Simple Models for Integrating Remotely Sensed Data With Ground Data

This section considers the use of relatively simple Bayesian models for combining two sources of data: multispectral data such as that obtained from Landsat TM data; and (categorical) ground data, for example soil maps produced from field surveys and/or photo interpretation. Of particular interest are the improvements (if any) in the classification accuracies which result from using the models and also any improvement gained by using an EM algorithm to estimate model parameters, as opposed to the *obvious* approach given by equation 4.4.

The *new* models presented in section 6.3 were jointly conceived, and some of the results presented here subsequently published, by Hojsgaard et al. (1997).

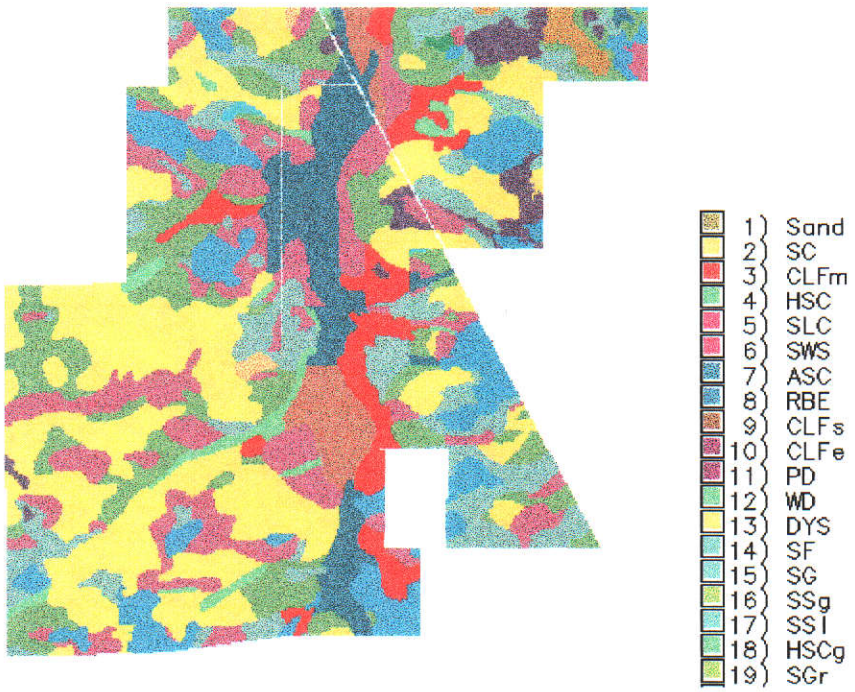



Figure 6.1: Image of soil types.

### 6.3.1 Data

#### Ground Data

Three sources of ground data were used in the experiments: soil type data, ground truth salinity data and paddock cropping records.

The soil type data was obtained from a local farming group, and a raster soil map was created (see figure 6.1). The soil map has a 30m pixel resolution to match that of the Landsat data. Nineteen different soil types are identified for the area, and are labelled $S = \{s_1, s_2, \ldots, s_{19}\}$ in the following discussion.

Detailed information on salinity was available from the farming group, in the form of a salinity *risk* map. The map was produced by the local landcare technician from aerial photo interpretation and field surveys. The categories identified were:

- SaltAffected – land that is presently considered to be affected by salinity.
- RiskOfSalinity – land that is considered not presently, but potentially, affected by salinity.

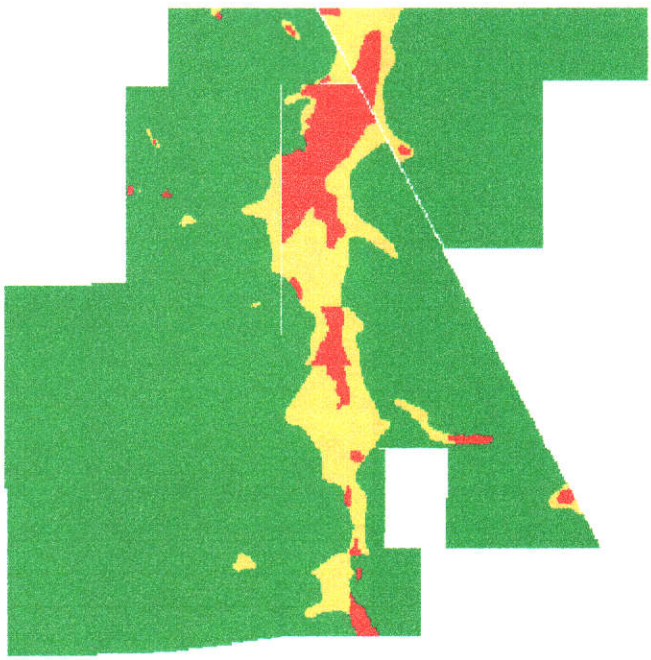


Figure 6.2: Salinity ground-truth: SaltAffected (red), RiskOfSalinity (yellow) and NoRisk (green).

- NoRiskOfSalinity – land not salt-affected and not likely to be affected in the future.

A raster map having a 30m pixel resolution was generated, and is shown in figure 6.2.

Farm paddock cropping histories were available for a number of paddocks within the area considered. These records were collected from farmers and identify, via *mud maps*[1], paddock boundaries and the paddock cropping history. Typically the landcover type is only known in broad terms; for example, knowledge that a paddock contains wheat is recorded, but not whether the wheat is in good or poor condition. These records were used to identify paddocks from the Landsat TM data from which areas of known crop type were digitised for use as ground truth. The paddock landuse types recorded were *Wheat*, *Lupins* and *Pasture*.

[1] A *mud map* is a crudely drawn sketch, typically not to scale.

#### Landsat TM Data

Landsat TM data, which can be processed to give information on landcovers, was available for the years 1989, 1990, 1991 and 1992. A single spring image was used for each of the years.

### 6.3.2 Existing Models

#### MLC

The Landsat TM data were separately classified for each of the years using MLC to produce land use/condition classifications having the following classes (described in section 3.2.2): *GoodCrop*, *PoorCrop*, *Lupins*, *GoodPasture*, *PoorPasture*, *PastureLikeSalt*, *SaltLikePasture*, *SaltMarginal*, *SaltSure*, *Remnant Vegetation* and *Water*. These are labelled $l^* = \{l_1^*, l_2^*, \ldots, l_{11}^*\}$ respectively in the following tables, where it will be assumed that the *true* land covers are represented by the variable $l = \{l_1, l_2, \ldots, l_{11}\} : l_i = l_i^*; i = 1 \ldots 11$. Figure 6.3 shows the class legends for the classifications discussed/used in this section; for convenience, classifications produced for the year 1990 will be used for illustration purposes. For visual comparison with the classifications produced by subsequent models, the MLC for 1990 is given in figure 6.3.

If the Landsat TM data are represented by the variable $y$, a MLC corresponds to calculating

$$p(l^*|y) = \frac{p(y|l^*)\, p(l^*)}{\sum_{l^*} p(y|l^*)\, p(l^*)} \tag{6.1}$$

The graphical representation for this model is depicted in figure 6.4(a). For this simple model, the representation simply indicates that the distributions $p(y|l^*)$ and $p(l^*)$ need to be specified.

#### MLC2

This model is based on the MLC and the availability of the MLC classification error rates. The classification errors for the MLC were estimated from landcover

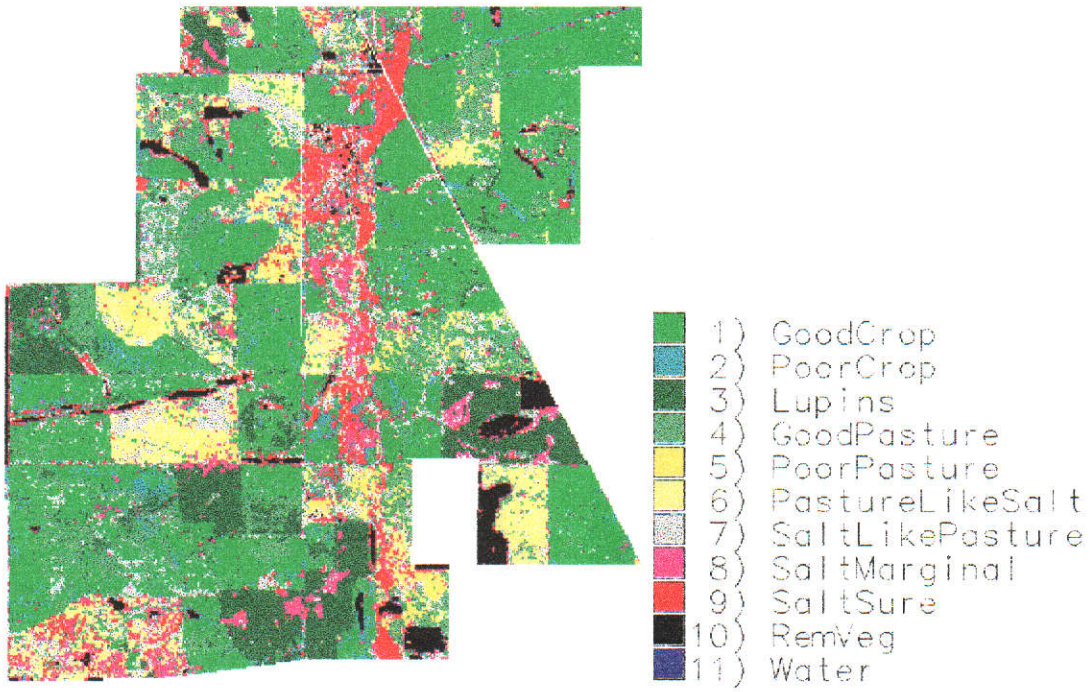



Figure 6.3: Classification produced by MLC.

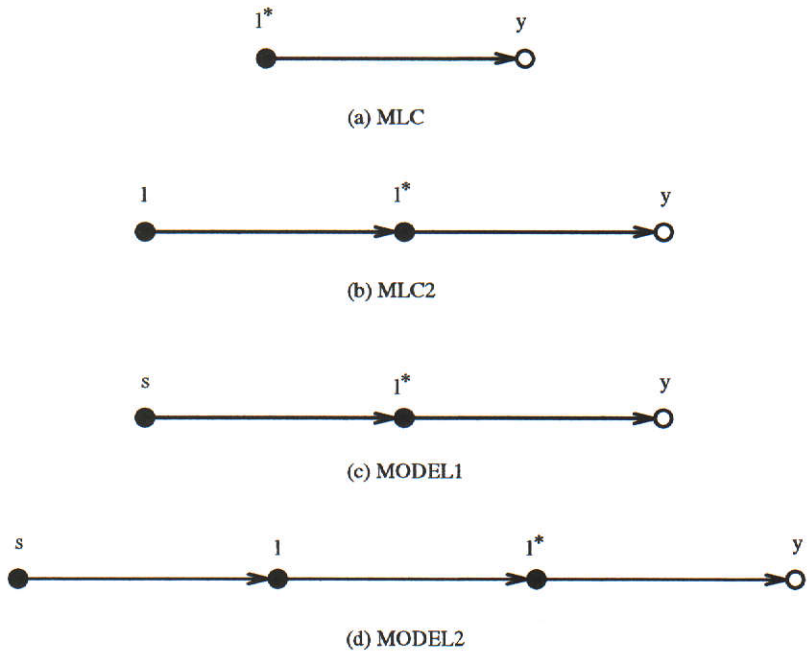



Figure 6.4: Simple Models: (a) MLC; (b) MLC2; (c) MODEL1; and (d) MODEL2.

| Year | Landcover Classes | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $l_1^*$ | $l_2^*$ | $l_3^*$ | $l_4^*$ | $l_5^*$ | $l_6^*$ | $l_7^*$ | $l_8^*$ | $l_9^*$ | $l_{10}^*$ | $l_{11}^*$ |
| 1989 | 0.65 | 0.65 | 0.42 | 0.69 | 0.69 | 0.69 | 0.88 | 0.88 | 0.88 | 0.90 | 1.00 |
| 1990 | 0.86 | 0.86 | 0.71 | 0.76 | 0.76 | 0.76 | 0.91 | 0.91 | 0.91 | 0.97 | 1.00 |
| 1991 | 0.70 | 0.70 | 0.20 | 0.66 | 0.66 | 0.66 | 0.84 | 0.84 | 0.84 | 0.99 | 1.00 |
| 1992 | 0.83 | 0.83 | 0.46 | 0.49 | 0.49 | 0.49 | 0.91 | 0.91 | 0.91 | 0.98 | 1.00 |

Table 6.1: Diagonal classification error estimate summary, $p(l_i^*|l_i)$, for each year $i$.

validation data covering areas of known landcovers (Furby, 1994; Furby et al., 1995), and were then summarised into the conditional probability table $p(l^*|l)$. This validation data consisted of many small sites scattered over an area much larger than that for which soils data were available, and hence could only be used for validating the spectral component of the models. The error-rates are given in table 6.1, where, for simplicity of the table, only the values $p(l_i^*|l_i)$ representing the diagonal entries of the classification confusion matrix are given.

The Landsat TM data were then processed using the model (depicted in figure 6.4(b) )

$$p(l|y) = \sum_{l^*} \frac{p(y|l^*)\, p(l^*|l)\, p(l)}{\sum_{l,l^*} p(y|l^*)\, p(l^*|l)\, p(l)} \tag{6.2}$$

which represents applying the MLC, and before forming class labels, updating the results using the classification error tables. The expectation here is that incorporating the error rates will reduce the number of misclassified pixels by *adjusting* the results from the spectral component of the model, in the sense that class membership probabilities are changed according to the MLC error estimates, before forming final class labels. The model relies on classification error rates being estimated as part of a MLC process. The rationale for incorporating these error rates explicitly in the model is:

> Given that a MLC has been performed, and the error rates calculated, then the variable $l$ should more closely represent the true land cover classes than the variable $l^*$. That is, $l^*$ repressents the classes defined for the (spectral) classifier, $l$ represents the true land classes and $p(l^*|l)$

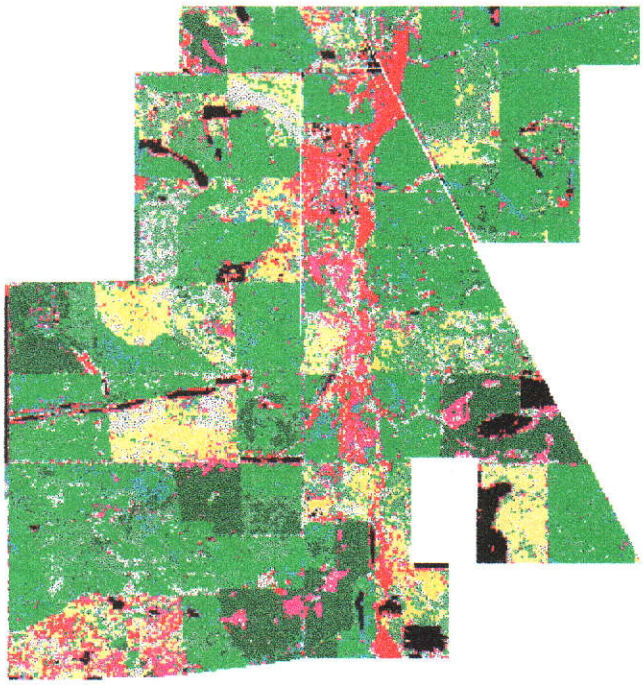


Figure 6.5: Classification produced by MLC2. The colour key is given in figure 6.3.

summarises the degree to which the spectral classes agree with the true landcover classes.

This model will be referred to as MLC2 in the following. An example of the classification is given in figure 6.5. Comparing figure 6.5 with figure 6.3, we observe that MLC2 produces an almost identical classification to MLC. However, the posterior probabilities associated with the class labels differs for the two models. This is expected to have an effect when combining multiple sources of data, as each data source can be weighted depending upon its accuracy. This will be examined further in the following sections.

### Multispectral Data and Ground Data – MODEL1

This model combines one source of multispectral data with one source of categorical data. In the following example, the source of multispectral data used is Landsat TM and a soil map provides the source of ground data, although other sources of each type of data could be combined using this scheme. This model,

called MODEL1 in the following discusion, assumes that the satellite data are independent of the soil data given the classification labels, and may be expressed mathematically as

$$p(l^*|y,s) = \frac{p(y|l^*)\ p(l^*|s)\ p(s)}{\sum_{l^*} p(y|l^*)\ p(l^*|s)\ p(s)} \tag{6.3}$$

and represented graphically as in figure 6.4(c).

MODEL1 uses the categorical data to influence the (prior) probability of observing a particular class label, $l^*$, and was first proposed by Strahler (1980). Other authors have used this general form of model in other applications. For example, Janssen and Middelkoop (1992) used knowledge of crop rotations and historical classifications to influence the probability of observing a crop in the *present* time period.

### 6.3.3 New Models

The models considered here are new in the following context: error rates of the spectral component of the classification process are explicitly included, and the EM algorithm is used for parameter estimation. In this way, the EM algorithm is integral to the approach.

#### MODEL2

The model considered here combines the ideas of MODEL1 and MLC2; that is, categorical data are used to influence the labelling of pixels (MODEL1), where in this case the results of the spectral component of the model are updated with estimates of the MLC error rates (MLC2). The idea behind this model is that categorical data are used to influence the variable representing the *true* landcover and hence improve classification accuracies and also yield estimates of $p(l|s)$ which should better represent the true relationship between soil and landcover than say $p(l^*|s)$ as used in MODEL1.

The model is expressed mathematically as:

$$p(l|y,s) = \sum_{l^*} \frac{p(y|l^*)\ p(l^*|l)\ p(l|s)\ p(s)}{\sum_{l,l^*} p(y|l^*)\ p(l^*|l)\ p(l|s)\ p(s)} \tag{6.4}$$

and is depicted graphically in figure 6.4(d). This model will be referred to as MODEL2 in the following.

### Incorporating the EM algorithm, MODEL1-EM and MODEL2-EM

For both MODEL1 and MODEL2, a conditional probability of a label given the ground data, that is $p(l^*|s)$ for MODEL1 and $p(l|s)$ for MODEL2, is needed. For convenience in the following discussion, the * will be dropped from $l^*$ where it is clear that $p(l^*|s)$ applies to MODEL1 and $p(l|s)$ to MODEL2. The quantity $p(l|s)$ can be established in a number of ways. An obvious method is simply to assume that the MLC labels are correct, in which case the following approach, which is simply equation 4.4 stated for this model, may be employed (termed OBVIOUS, or OBV for short, in the following):

> Using the labels produced by the MLC, estimate the parameters $p(l|s)$ by calculating $p(l|s) = n(l,s)/n(s)$, where $n(s)$ denotes the number of times each class of soil is observed and $n(l,s)$ denotes the number of times a land cover class and a soil type class are jointly observed.

The parameters for $p(l|s)$ were estimated using this technique and are given, for comparison with parameters derived using an EM algorithm, in table 6.2.

The data were then processed with MODEL1 and MODEL2 and the EM algorithm was used to estimate the parameters. In the following, the simpler of these models (equation 6.3) will be referred to as MODEL1-EM and the model with two latent variables (equation 6.4) will be referred to as MODEL2-EM.

Of interest was the benefit that could be gained by including the ground data and by estimating the parameters using an EM algorithm, as opposed to some obvious method, for example the OBVIOUS approach presented above. Uniform

| | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ | $l_6$ | $l_7$ | $l_8$ | $l_9$ | $l_{10}$ | $l_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | 0.28 | 0.10 | 0.08 | 0.05 | 0.05 | 0.21 | 0.07 | 0.06 | 0.04 | 0.06 | 0.00 |
| $s_2$ | 0.37 | 0.01 | 0.25 | 0.18 | 0.00 | 0.06 | 0.06 | 0.07 | 0.01 | 0.00 | 0.00 |
| $s_3$ | 0.73 | 0.08 | 0.00 | 0.03 | 0.01 | 0.02 | 0.04 | 0.03 | 0.05 | 0.00 | 0.00 |
| $s_4$ | 0.63 | 0.04 | 0.24 | 0.02 | 0.00 | 0.01 | 0.02 | 0.03 | 0.01 | 0.00 | 0.00 |
| $s_5$ | 0.45 | 0.04 | 0.05 | 0.17 | 0.03 | 0.08 | 0.08 | 0.05 | 0.04 | 0.00 | 0.00 |
| $s_6$ | 0.22 | 0.04 | 0.10 | 0.13 | 0.06 | 0.16 | 0.12 | 0.07 | 0.07 | 0.02 | 0.00 |
| $s_7$ | 0.10 | 0.04 | 0.02 | 0.04 | 0.02 | 0.05 | 0.06 | 0.11 | 0.53 | 0.03 | 0.01 |
| $s_8$ | 0.46 | 0.09 | 0.03 | 0.08 | 0.01 | 0.17 | 0.08 | 0.05 | 0.02 | 0.02 | 0.00 |
| $s_9$ | 0.35 | 0.06 | 0.02 | 0.20 | 0.03 | 0.07 | 0.10 | 0.09 | 0.08 | 0.00 | 0.00 |
| $s_{10}$ | 0.29 | 0.08 | 0.01 | 0.14 | 0.02 | 0.07 | 0.08 | 0.15 | 0.13 | 0.04 | 0.00 |
| $s_{11}$ | 0.34 | 0.09 | 0.07 | 0.10 | 0.03 | 0.12 | 0.10 | 0.07 | 0.05 | 0.03 | 0.00 |
| $s_{12}$ | 0.15 | 0.02 | 0.15 | 0.07 | 0.03 | 0.33 | 0.10 | 0.04 | 0.04 | 0.01 | 0.00 |
| $s_{13}$ | 0.37 | 0.03 | 0.11 | 0.11 | 0.02 | 0.16 | 0.12 | 0.03 | 0.03 | 0.03 | 0.00 |
| $s_{14}$ | 0.01 | 0.03 | 0.00 | 0.01 | 0.04 | 0.11 | 0.10 | 0.14 | 0.52 | 0.05 | 0.00 |
| $s_{15}$ | 0.36 | 0.04 | 0.09 | 0.08 | 0.03 | 0.17 | 0.11 | 0.04 | 0.04 | 0.02 | 0.00 |
| $s_{16}$ | 0.28 | 0.06 | 0.03 | 0.10 | 0.04 | 0.14 | 0.09 | 0.10 | 0.05 | 0.11 | 0.00 |
| $s_{17}$ | 0.36 | 0.06 | 0.05 | 0.09 | 0.02 | 0.14 | 0.09 | 0.05 | 0.07 | 0.05 | 0.00 |
| $s_{18}$ | 0.08 | 0.00 | 0.00 | 0.69 | 0.04 | 0.00 | 0.04 | 0.04 | 0.12 | 0.00 | 0.00 |
| $s_{19}$ | 0.05 | 0.05 | 0.01 | 0.05 | 0.00 | 0.24 | 0.54 | 0.01 | 0.04 | 0.00 | 0.00 |

Table 6.2: Parameter estimates derived by OBVIOUS means.

distributions for $p(l|s)$ were specified as the initial parameter estimates in the EM algorithms.

The estimated parameters $p(l|s)$ for MODEL1-EM are given in table 6.5 and for MODEL2–EM in table 6.6. In all cases, 50 iterations of the EM algorithm were specified.

Typically, EM algorithms require many iterations to converge, but for some models, near optimal results may be obtained after a few iterations. Table 6.3 and table 6.4 summarise the convergence properties for the first 10 iterations of the algorithm, from which it is observed that little (relative) change occurs in the parameter estimates after about the fifth or sixth iteration. The estimates obtained are given in tables 6.5 and 6.6, and the classifications are depicted in figures 6.6 and 6.7.

| Iteration | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| MaxDev 1989 | 0.91 | 0.20 | 0.08 | 0.04 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 |
| MaxDev 1990 | 0.91 | 0.10 | 0.03 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| MaxDev 1991 | 0.91 | 0.12 | 0.07 | 0.04 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 |
| MaxDev 1992 | 0.91 | 0.14 | 0.03 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Table 6.3: MODEL1-EM parameter convergence.

| Iteration | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| MaxDev 1989 | 0.91 | 0.20 | 0.15 | 0.10 | 0.06 | 0.04 | 0.03 | 0.02 | 0.01 | 0.01 |
| MaxDev 1990 | 0.91 | 0.22 | 0.09 | 0.04 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 |
| MaxDev 1991 | 0.91 | 0.20 | 0.12 | 0.07 | 0.04 | 0.03 | 0.02 | 0.02 | 0.02 | 0.02 |
| MaxDev 1992 | 0.91 | 0.25 | 0.12 | 0.05 | 0.03 | 0.02 | 0.01 | 0.01 | 0.01 | 0.00 |

Table 6.4: MODEL2-EM parameter convergence.

| | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ | $l_6$ | $l_7$ | $l_8$ | $l_9$ | $l_{10}$ | $l_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | 0.31 | 0.09 | 0.08 | 0.03 | 0.04 | 0.24 | 0.06 | 0.06 | 0.04 | 0.06 | 0.00 |
| $s_2$ | 0.40 | 0.00 | 0.25 | 0.18 | 0.00 | 0.05 | 0.05 | 0.06 | 0.00 | 0.00 | 0.00 |
| $s_3$ | 0.80 | 0.05 | 0.00 | 0.13 | 0.00 | 0.01 | 0.03 | 0.02 | 0.05 | 0.00 | 0.01 |
| $s_4$ | 0.68 | 0.02 | 0.24 | 0.01 | 0.00 | 0.01 | 0.01 | 0.03 | 0.01 | 0.00 | 0.00 |
| $s_5$ | 0.51 | 0.03 | 0.04 | 0.16 | 0.02 | 0.08 | 0.08 | 0.04 | 0.03 | 0.00 | 0.00 |
| $s_6$ | 0.24 | 0.02 | 0.10 | 0.14 | 0.05 | 0.18 | 0.13 | 0.06 | 0.06 | 0.02 | 0.00 |
| $s_7$ | 0.11 | 0.03 | 0.01 | 0.03 | 0.02 | 0.05 | 0.04 | 0.09 | 0.59 | 0.02 | 0.00 |
| $s_8$ | 0.52 | 0.07 | 0.03 | 0.06 | 0.00 | 0.18 | 0.08 | 0.03 | 0.02 | 0.02 | 0.00 |
| $s_9$ | 0.39 | 0.04 | 0.01 | 0.22 | 0.03 | 0.06 | 0.09 | 0.09 | 0.08 | 0.00 | 0.00 |
| $s_{10}$ | 0.31 | 0.07 | 0.01 | 0.14 | 0.01 | 0.05 | 0.07 | 0.17 | 0.13 | 0.03 | 0.00 |
| $s_{11}$ | 0.38 | 0.08 | 0.07 | 0.09 | 0.02 | 0.13 | 0.10 | 0.07 | 0.04 | 0.03 | 0.00 |
| $s_{12}$ | 0.15 | 0.00 | 0.15 | 0.07 | 0.01 | 0.37 | 0.18 | 0.03 | 0.02 | 0.01 | 0.00 |
| $s_{13}$ | 0.42 | 0.01 | 0.11 | 0.09 | 0.01 | 0.16 | 0.13 | 0.02 | 0.02 | 0.02 | 0.00 |
| $s_{14}$ | 0.01 | 0.01 | 0.00 | 0.00 | 0.02 | 0.13 | 0.07 | 0.12 | 0.59 | 0.04 | 0.00 |
| $s_{15}$ | 0.40 | 0.03 | 0.09 | 0.07 | 0.02 | 0.19 | 0.11 | 0.03 | 0.03 | 0.02 | 0.00 |
| $s_{16}$ | 0.31 | 0.05 | 0.03 | 0.09 | 0.03 | 0.15 | 0.09 | 0.10 | 0.04 | 0.11 | 0.00 |
| $s_{17}$ | 0.40 | 0.05 | 0.05 | 0.08 | 0.01 | 0.15 | 0.09 | 0.04 | 0.08 | 0.05 | 0.00 |
| $s_{18}$ | 0.04 | 0.00 | 0.00 | 0.80 | 0.00 | 0.00 | 0.00 | 0.02 | 0.14 | 0.00 | 0.00 |
| $s_{19}$ | 0.05 | 0.05 | 0.00 | 0.03 | 0.00 | 0.21 | 0.64 | 0.00 | 0.01 | 0.00 | 0.00 |

Table 6.5: MODEL1–EM parameter estimates.

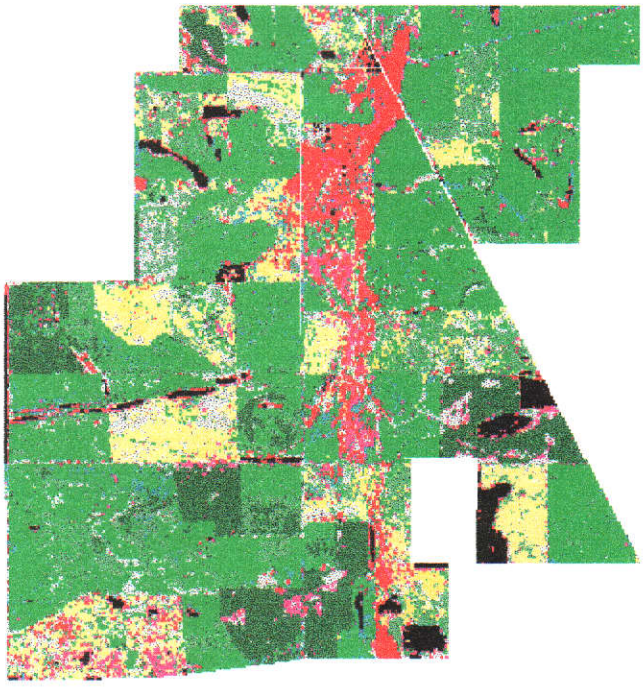


Figure 6.6: MODEL1–EM classification. The colour key is given in figure 6.3.

| | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ | $l_6$ | $l_7$ | $l_8$ | $l_9$ | $l_{10}$ | $l_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | 0.39 | 0.10 | 0.11 | 0.00 | 0.04 | 0.27 | 0.00 | 0.00 | 0.01 | 0.06 | 0.00 |
| $s_2$ | 0.43 | 0.00 | 0.41 | 0.13 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 |
| $s_3$ | 0.89 | 0.05 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 |
| $s_4$ | 0.62 | 0.00 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $s_5$ | 0.72 | 0.02 | 0.04 | 0.17 | 0.01 | 0.01 | 0.02 | 0.00 | 0.01 | 0.00 | 0.00 |
| $s_6$ | 0.28 | 0.00 | 0.17 | 0.15 | 0.07 | 0.22 | 0.04 | 0.00 | 0.04 | 0.02 | 0.00 |
| $s_7$ | 0.13 | 0.02 | 0.02 | 0.01 | 0.02 | 0.01 | 0.03 | 0.09 | 0.65 | 0.01 | 0.00 |
| $s_8$ | 0.73 | 0.09 | 0.01 | 0.02 | 0.00 | 0.12 | 0.01 | 0.00 | 0.00 | 0.02 | 0.00 |
| $s_9$ | 0.53 | 0.04 | 0.00 | 0.31 | 0.02 | 0.01 | 0.01 | 0.02 | 0.05 | 0.00 | 0.00 |
| $s_{10}$ | 0.42 | 0.09 | 0.00 | 0.18 | 0.00 | 0.00 | 0.01 | 0.14 | 0.13 | 0.03 | 0.00 |
| $s_{11}$ | 0.52 | 0.10 | 0.10 | 0.07 | 0.01 | 0.10 | 0.04 | 0.02 | 0.02 | 0.03 | 0.00 |
| $s_{12}$ | 0.13 | 0.00 | 0.25 | 0.02 | 0.01 | 0.48 | 0.10 | 0.00 | 0.00 | 0.01 | 0.00 |
| $s_{13}$ | 0.55 | 0.00 | 0.16 | 0.04 | 0.00 | 0.13 | 0.08 | 0.00 | 0.01 | 0.02 | 0.00 |
| $s_{14}$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.16 | 0.04 | 0.10 | 0.62 | 0.03 | 0.00 |
| $s_{15}$ | 0.54 | 0.01 | 0.13 | 0.02 | 0.02 | 0.19 | 0.05 | 0.00 | 0.01 | 0.02 | 0.00 |
| $s_{16}$ | 0.44 | 0.06 | 0.02 | 0.10 | 0.03 | 0.15 | 0.02 | 0.05 | 0.02 | 0.11 | 0.00 |
| $s_{17}$ | 0.56 | 0.05 | 0.06 | 0.06 | 0.00 | 0.12 | 0.03 | 0.00 | 0.06 | 0.05 | 0.00 |
| $s_{18}$ | 0.05 | 0.00 | 0.00 | 0.89 | 0.00 | 0.00 | 0.00 | 0.00 | 0.06 | 0.00 | 0.00 |
| $s_{19}$ | 0.06 | 0.04 | 0.00 | 0.00 | 0.00 | 0.23 | 0.67 | 0.00 | 0.00 | 0.00 | 0.00 |

Table 6.6: MODEL2–EM parameter estimates.

of commonality is that the SaltSure class as defined for the Bayesian models is a subset of the technicians' SaltAffected class. For this reason, the following rationale was used to judge the salinity mapping performance of the classifiers:

> We can be confident that the technicians NoRisk class should have no SaltSure or SaltMarginal areas in it, whereas the technicians' RiskOf-Salinity and SaltAffected classes may or may not have SaltSure and SaltMarginal areas in them.

Hence the performance of each model was judged relative to how many times it falsely labelled SaltSure and SaltMarginal areas in NoRisk areas. The results are given in table 6.7, where the numbers represent the total number of pixels identified as SaltSure and SaltMarginal (out of a total of 306550 pixels), and the percentages reflect the proportions of these labels that are considered incorrect by the above criterion.

From the table, we observe that the four classifiers identify roughly the same number of SaltSure pixels, with MODEL1–EM reducing the total number of misclassified pixels by 2% and MODEL2–EM by 5% as compared with the MLC. For the SaltMarginal class, MODEL1–EM reduces the total number of misclassified pixels by 15% and MODEL2–EM by 19% as compared to the MLC. MODEL2–EM identified a lot less pixels as having this class. For the two classes combined, MODEL1–EM and MODEL2–EM produce 4% and 8% less errors respectively than the MLC.

From table 6.7, it appears that using the classification error rates in conjunction with soil data is advantageous. A question that immediately arises is: what benefit is the EM algorithm over an obvious estimation technique which requires less computational effort. To examine this, the data were processed with MODEL2 using the parameter estimates given in table 6.2. Note that this approach is not iterative. This model will be referred to as MODEL2–OBV. This approach is somewhat dubious in this case, as we are in a sense using the information twice. However, we could imagine that reliable estimates for $p(l|s)$ were available from say

| Salt Class | Model / Classifier | | | | |
|---|---|---|---|---|---|
| | MLC | MLC2 | MODEL1-EM | MODEL2-EM | MODEL2-OBV |
| | Year 1989 | | | | |
| $\|SaltSure\|$ | 46770 | 47871 | 57827 | 59736 | 53903 |
| $\|SaltSure_{err}\|$ | 22917 | 23457 | 26600 | 25686 | 24795 |
| % *Incorrect* | 49 | 49 | 46 | 43 | 46 |
| $\|SaltMarginal\|$ | 25255 | 26940 | 14073 | 5864 | 17767 |
| $\|SaltMarginal_{err}\|$ | 16416 | 17780 | 8584 | 2111 | 10660 |
| % *Incorrect* | 65 | 66 | 61 | 36 | 60 |
| | Year 1990 | | | | |
| $\|SaltSure\|$ | 63978 | 64441 | 73144 | 72071 | 71943 |
| $\|SaltSure_{err}\|$ | 33269 | 33509 | 38766 | 37477 | 37410 |
| % *Incorrect* | 52 | 52 | 53 | 52 | 52 |
| $\|SaltMarginal\|$ | 19933 | 20335 | 14310 | 11581 | 15761 |
| $\|SaltMarginal_{err}\|$ | 11561 | 10981 | 8014 | 5791 | 8984 |
| % *Incorrect* | 58 | 54 | 56 | 50 | 57 |
| | Year 1991 | | | | |
| $\|SaltSure\|$ | 30754 | 31651 | 38226 | 37040 | 35085 |
| $\|SaltSure_{err}\|$ | 11994 | 12344 | 12997 | 10742 | 12280 |
| % *Incorrect* | 39 | 39 | 34 | 29 | 35 |
| $\|SaltMarginal\|$ | 15487 | 15612 | 11572 | 11241 | 13241 |
| $\|SaltMarginal_{err}\|$ | 4181 | 4215 | 1852 | 1461 | 2648 |
| % *Incorrect* | 27 | 27 | 16 | 13 | 20 |
| | Year 1992 | | | | |
| $\|SaltSure\|$ | 35346 | 35911 | 36119 | 36204 | 36558 |
| $\|SaltSure_{err}\|$ | 10604 | 11132 | 9391 | 7065 | 9505 |
| % *Incorrect* | 30 | 31 | 26 | 22 | 26 |
| $\|SaltMarginal\|$ | 22321 | 23287 | 20545 | 12198 | 20785 |
| $\|SaltMarginal_{err}\|$ | 12946 | 13506 | 11300 | 5001 | 11432 |
| % *Incorrect* | 58 | 58 | 55 | 41 | 55 |
| | Total-Errors, All Years | | | | |
| $\|SaltSure\|$ | 175076 | 178760 | 204081 | 204675 | 195328 |
| $\|SaltSure_{err}\|$ | 78784 | 80442 | 87755 | 81870 | 83991 |
| % *Incorrect* | 45 | 45 | 43 | 40 | 43 |
| $\|SaltMarginal\|$ | 83528 | 86080 | 60455 | 41040 | 67448 |
| $\|SaltMarginal_{err}\|$ | 45105 | 46483 | 29623 | 14364 | 33724 |
| % *Incorrect* | 54 | 54 | 49 | 35 | 50 |
| | Total-Errors, All Years/All Classes | | | | |
| $\|SaltAffected\|$ | 258604 | 264840 | 264536 | 245715 | 262776 |
| $\|SaltAffected_{err}\|$ | **123889** | **126925** | **117378** | **96234** | **117715** |
| **% Incorrect** | **48** | **48** | **44** | **40** | **45** |

Table 6.7: Model/classifier comparisons.

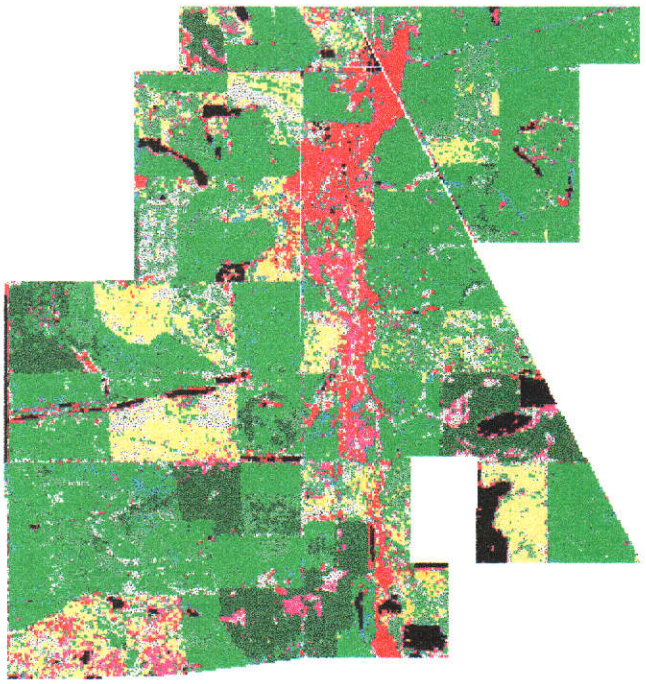


Figure 6.8: MODEL2–OBV classification output. The colour key is given in figure 6.3.

other areas similar to the study area or from other sources of prior information. The classification accuracy of MODEL2–OBV is summarised in table 6.7, and depicted in figure 6.8. From table 6.7, we observe that MODEL2–OBV performs similarly to MODEL1–EM, which is not as good as MODEL2–EM.

Finally, it is known that *SaltSure* areas should generally persist as *SaltSure* in each of the years. From table 6.7, we observe that for a given model, the estimates of $|SaltSure|$ vary significantly from year-to-year over the time period considered, which should not be the case. This is the result of bare areas in salt-prone soil types being correctly interpreted as *SaltSure*. From this, we conclude that although soil type (or indeed other ground data) may improve salinity mapping, models which include temporal constraints on the occurrence of salinity will improve the results. Temporal constraints are considered in section 6.4.

### Paddock-use mapping accuracies

Paddock-use mapping accuracies were estimated by comparing the various classifications with the available paddock cropping information. The paddock cropping information identifies three broad land-cover classes: *Wheat*, *Lupins* and *Pasture*. For this reason, the classification were grouped into the broad classes:

- $Wheat = \{GoodCrop,\ PoorCrop\}$
- $Lupins$
- $Pasture = \{GoodPasture,\ PoorPasture,\ PastureLikeSalt\}$
- $Other = \{SaltLikePasture,\ SaltMarginal,\ SaltSure,\ RemVeg,\ Water\}$

for comparison with the ground truth information. The classification confusion matrices for the years 1989, 1990, 1991 and 1992 are given in tables 6.8, 6.9, 6.10 and 6.11 respectively. The average performance of the classifiers is summarised in table 6.12. Here, the average performance is considered to be the total number of correctly labelled pixels over the time period 1989 to 1992, expressed as a percentage of the total pixels classified. From the tables, it is observed that the MLC, MLC2, MODEL1–EM and MODEL2–OBV perform similarly, whereas MODEL2–EM typically has poorer performance (an average of 7% less accurate than the MLC).

The estimates $p(l|s)$ provide some insight into the relative performance of the models. For example, in 1992 the estimates for $p(l_1|s_3)$ from MODEL2–OBV, MODEL1–EM and MODEL2–EM (tables 6.2, 6.5 and 6.6 respectively) are 0.73, 0.80 and 0.89 respectively, indicating that the soil information has a greater influence on the outcome of $l_1$ for MODEL2–EM than for either MODEL2–OBV or MODEL1–EM. Indeed, for each class of soil, the magnitude of the estimated parameter associated with the most likely class is greater for MODEL2–EM than for the other two models, indicating that MODEL2–EM makes *more* use of the local uniformity of the ground data. Comparing the classification performance of

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | **% Correct** |
| | MLC | | | | **Average 60** |
| Wheat | 5406 | 3 | 3047 | 1757 | 53 |
| Lupins | 31 | 147 | 212 | 50 | 33 |
| Pasture | 404 | 130 | 3804 | 1055 | 71 |
| | MLC2 | | | | **Average 57** |
| Wheat | 5336 | 3 | 2958 | 1916 | 52 |
| Lupins | 32 | 145 | 210 | 53 | 33 |
| Pasture | 406 | 126 | 3696 | 1165 | 69 |
| | MODEL1-EM | | | | **Average 59** |
| Wheat | 5236 | 0 | 3845 | 1132 | 51 |
| Lupins | 32 | 116 | 283 | 9 | 26 |
| Pasture | 344 | 63 | 4097 | 889 | 76 |
| | MODEL2-EM | | | | **Average 54** |
| Wheat | 4582 | 0 | 4863 | 768 | 45 |
| Lupins | 23 | 116 | 300 | 1 | 26 |
| Pasture | 489 | 60 | 3978 | 866 | 74 |
| | MODEL2-OBV | | | | **Average 58** |
| Wheat | 5293 | 0 | 3576 | 1344 | 52 |
| Lupins | 35 | 115 | 274 | 16 | 26 |
| Pasture | 383 | 62 | 3978 | 970 | 74 |

Table 6.8: Paddock-use mapping accuracies, 1989.

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | % Correct |
| | MLC | | | | **Average 73** |
| Wheat | 4133 | 7 | 312 | 579 | 82 |
| Lupins | 78 | 1757 | 581 | 173 | 68 |
| Pasture | 306 | 110 | 5647 | 2031 | 70 |
| | MLC2 | | | | **Average 73** |
| Wheat | 4133 | 7 | 307 | 584 | 82 |
| Lupins | 80 | 1752 | 583 | 174 | 68 |
| Pasture | 313 | 110 | 5601 | 2070 | 69 |
| | MODEL1-EM | | | | **Average 74** |
| Wheat | 4220 | 2 | 314 | 495 | 84 |
| Lupins | 97 | 1740 | 537 | 215 | 67 |
| Pasture | 393 | 63 | 5616 | 2022 | 69 |
| | MODEL2-EM | | | | **Average 73** |
| Wheat | 4284 | 1 | 374 | 372 | 85 |
| Lupins | 128 | 1736 | 508 | 217 | 67 |
| Pasture | 734 | 46 | 5464 | 1850 | 68 |
| | MODEL2-OBV | | | | **Average 74** |
| Wheat | 4245 | 3 | 308 | 475 | 84 |
| Lupins | 104 | 1730 | 541 | 214 | 67 |
| Pasture | 422 | 63 | 5593 | 2016 | 69 |

Table 6.9: Paddock-use mapping accuracies, 1990.

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | % Correct |
| | MLC | | | | **Average 71** |
| Wheat | 5683 | 39 | 465 | 105 | 90 |
| Lupins | 80 | 1150 | 139 | 3 | 84 |
| Pasture | 2780 | 246 | 4638 | 718 | 55 |
| | MLC2 | | | | **Average 70** |
| Wheat | 5680 | 8 | 483 | 121 | 90 |
| Lupins | 191 | 897 | 280 | 4 | 65 |
| Pasture | 2804 | 107 | 4679 | 792 | 56 |
| | MODEL1-EM | | | | **Average 69** |
| Wheat | 5921 | 4 | 268 | 99 | 94 |
| Lupins | 239 | 852 | 280 | 1 | 62 |
| Pasture | 3273 | 79 | 4223 | 807 | 50 |
| | MODEL2-EM | | | | **Average 56** |
| Wheat | 5573 | 0 | 595 | 124 | 86 |
| Lupins | 795 | 0 | 576 | 1 | 0 |
| Pasture | 4325 | 0 | 3420 | 637 | 41 |
| | MODEL2-OBV | | | | **Average 64** |
| Wheat | 5937 | 1 | 258 | 96 | 94 |
| Lupins | 763 | 222 | 386 | 1 | 16 |
| Pasture | 3414 | 44 | 4175 | 749 | 50 |

Table 6.10: Paddock-use mapping accuracies, 1991.

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | % Correct |
| | MLC | | | | **Average 85** |
| Wheat | 6610 | 62 | 530 | 192 | 89 |
| Lupins | 0 | 482 | 1 | 5 | 99 |
| Pasture | 370 | 63 | 6548 | 1183 | 80 |
| | MLC2 | | | | **Average 85** |
| Wheat | 6719 | 61 | 419 | 195 | 91 |
| Lupins | 0 | 482 | 1 | 5 | 99 |
| Pasture | 415 | 62 | 6428 | 1259 | 79 |
| | MODEL1-EM | | | | **Average 86** |
| Wheat | 6856 | 51 | 327 | 160 | 93 |
| Lupins | 0 | 482 | 1 | 5 | 99 |
| Pasture | 458 | 60 | 6518 | 1128 | 80 |
| | MODEL2-EM | | | | **Average 77** |
| Wheat | 7086 | 35 | 231 | 42 | 96 |
| Lupins | 221 | 263 | 3 | 1 | 54 |
| Pasture | 2006 | 52 | 5016 | 1090 | 61 |
| | MODEL2-OBV | | | | **Average 84** |
| Wheat | 6947 | 34 | 251 | 162 | 94 |
| Lupins | 0 | 481 | 1 | 6 | 99 |
| Pasture | 659 | 51 | 6238 | 1216 | 76 |

Table 6.11: Paddock-use mapping accuracies, 1992.

| MLC | MLC2 | MODEL1-EM | MODEL2-EM | MODEL2-OBV |
|---|---|---|---|---|
| 72 | 71 | 72 | 65 | 70 |

Table 6.12: Average paddock-use accuracies (%), 1989–1992.

MODEL2–EM with MODEL1–EM, this is not surprising as incorporating $p(l^*|l)$ would tend to reduce the magnitude of the evidence presented from the spectral data, allowing the soils data to have greater potential for changing the resulting class label. Comparing the classification performance of MODEL2–EM with MODEL2–OBV, it is evident that enough pixels changed labels such that iteratively updating the parameters had a significant effect on the label outcomes.

The increased influence that MODEL2–EM allowed the ground data to have on the resulting class label resulted in this model outperforming the other models for the salinity criteria but performing the worst for crop mapping. This is largely due to the following:

- Salinity is strongly associated with soils such as *ClayFlat* but rarely occurs on soils such as *DeepSand*. In this way, the soil data provide a clear partitioning for what is likely to be saline or not and hence the model performs well when judged by this criteria.

- In this region, farmers do not tend to crop specifically according to soil type. Rather, other management considerations such as cropping rotations and the need for managing livestock (in which case existing fencing dominates the spatial characteristics of homogeneous land use regions). For this reason, the relative occurrence of crop varieties over a large number of soil types will tend to be incidental rather than planned. MODEL2–EM, which allows the soil information to have greater influence on the final labelling, tends to substitute the most probable crop varieties for a given soil type for the true crop variety. For example, MODEL2–EM maps a good proportion of lupins as wheat, as for a given soil type wheat is generally more probably than lupins.

For MODEL1–EM and MODEL2–OBV, more modest increases in the salinity mapping accuracies are achieved while crop mapping accuracies are similar to that of the MLC.

| | MLC2 | MODEL1–EM | MODEL2–EM | MODEL2–OBV |
|---|---|---|---|---|
| % Improvement 1989 | -1 | 15 | 26 | 11 |
| % Improvement 1990 | 0 | 8 | 11 | 7 |
| % Improvement 1991 | 1 | 10 | 26 | 12 |
| % Improvement 1992 | 1 | 9 | 26 | 10 |
| **% Average Improvement** | **0** | **10** | **22** | **10** |

Table 6.13: Summary of the spatial uniformity improvements obtained from each of the models.

### Spatial Uniformity

The spatial uniformity of the class labels produced by the different techniques can also be examined. If a label is different from a neighbouring label, then one count is added to the classification total uniformity measure. That is, the uniformity measure is the total number of pairs of adjacent pixels which have different class labels. For each of the models, a total uniformity measure of the resulting classification was made, and then this value was summarised as a percentage of the MLC uniformity measure. The results are presented in table 6.13. From table 6.13, we observe that the models which incorporate the soil data have greater spatial uniformity than those which do not, and hence provide some form of smoothing. This smoothing effect is the result of the soil data forming a relatively homogeneous partitioning, as compared to the size of a Landsat TM pixel, of the region. Of the models incorporating the soil data, MODEL2–EM produces classifications which consistently have significantly greater spatial uniformity than the other models, having on average 22% less pixels different from their neighbours than the MLC. This is not surprising given that the soil data have a greater opportunity in MODEL2–EM to influence the final labels.

## 6.3.5 Conclusions

For the simple models considered, using soil data to influence the prior probability during pixel labelling improves salinity mapping accuracies for each of the

models considered. Comparing MODEL1–EM with MODEL2–EM, these accuracies improved when error estimates of the spectral component of the process were included.

For paddock-use mapping, all models except MODEL2–EM performed similarly to the MLC, suggesting that soil data are of limited use for this purpose. MODEL2–EM performed the worst by this criteria, which is a consequence of this model allowing the soil data to have greater effect on the final pixel labelling as compared with the other models. This demonstrates that while it can be beneficial to combine ground data with remotely sensed data with the models considered, certain sources of ground data may in fact reduce the accuracy of certain classification tasks, for example using soil data for mapping different paddock uses. In this case the soil data may be better applied by forming a prior for the broad classes {SaltAffected, NotSaltAffected}, in which case no distinction between (non-salt affected) wheat and pasture, say, is made on the basis of soil type.

For estimating parameters in models with latent variables, the EM algorithm is appropriate. Comparing the results of MODEL2–EM with MODEL2–OBV, the iterative nature of the EM algorithm allows for greater smoothing as opposed to non-iterative methods. When the ground data captured the spatial correlation of the classification task, mapping what was salt-affected and what was not, the model based on EM outperformed the non-iterative approach. For all models incorporating the ground data, improvements in the spatial consistency of the resulting interpretations were obtained.

From experiments with the above simple models, the following factors appear advantageous when constructing Bayesian models for the interpretation of land-use/condition:

1. The use of appropriate ancillary data for a particular classification task.

2. Estimates of the errors associated with the variables considered.

3. The use of the EM algorithm for estimating model parameters.

It has been demonstrated that the incorporation of ancillary data can improve the accuracy of mapping salinity, but also that that accurate identification of salinity requires more than a single year of imagery. More complicated models are required to remove the mistakes that are obvious when the relative temporal stability of salinity is recognised. The effect of temporal rules is considered in section 6.4.

Finally, while it was observed that combining data has the ability to provide broad-scale smoothing, some spatial inconsistency in the resulting labels still remained. In section 6.4, a method which allows the spatial relationships of neighbouring pixels to be explicitly included in Bayesian networks is considered.

## 6.4 Simple Temporal and Spatial Models

In section 6.3, simple models that combined one source of multispectral imagery with one source of ground data were considered. While improvements in the accuracy of mapping salinity were obtained, it was observed that ancillary data alone could not remove areas falsely labelled as saline: it is likely that Landsat TM data (or other data) from more than one year are required.

In this section, obvious temporal models for combining single-date imagery from multiple years are considered. Ground data are explicitly ignored in these models, so that the effect of combining temporal Landsat TM data may be observed.

In particular, the models make the simple assumption that landcover as observed in any one year, year $t$ say, is conditionally independent of all other years given the previous $(t-1)$ and following $(t+1)$ year. It would be expected that the temporal rules will increase the classification accuracy of salinity and also of paddock-use.

In section 6.3, it was observed that the process of combining data provided some broad-scale smoothing effects, but that some errors (spatially inconsistent

labels) still remained. In this section, the broad-scale smoothing effects of the temporal rules are also examined. A means of providing small-scale smoothing is demonstated.

### 6.4.1 Temporal Models – MODEL3 and MODEL4

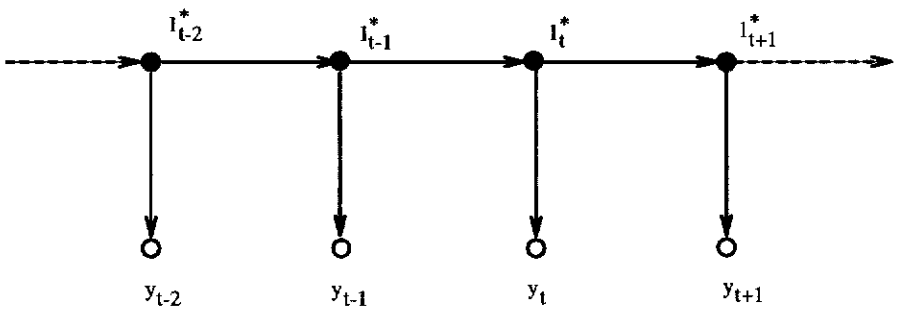
In the following, the variables $y$, $l$ and $l^*$ have the some meaning as in the previous section, except here they have the subscript $t$ to identify the year $t$. Such a model is depicted graphically in figure 6.9(a), and will be referred to as MODEL3 in the following. Drawing on the experiments of section 6.3, a model which incorporates the classification error rates is also proposed, and is depicted in figure 6.9(b). This model will be called MODEL4 in the following.

The four years of Landsat TM data considered in section 6.3 were combined using the models. That is, from the graphical depictions given in figure 6.9, only relationships represented by solid lines are present and $t = 1991$.

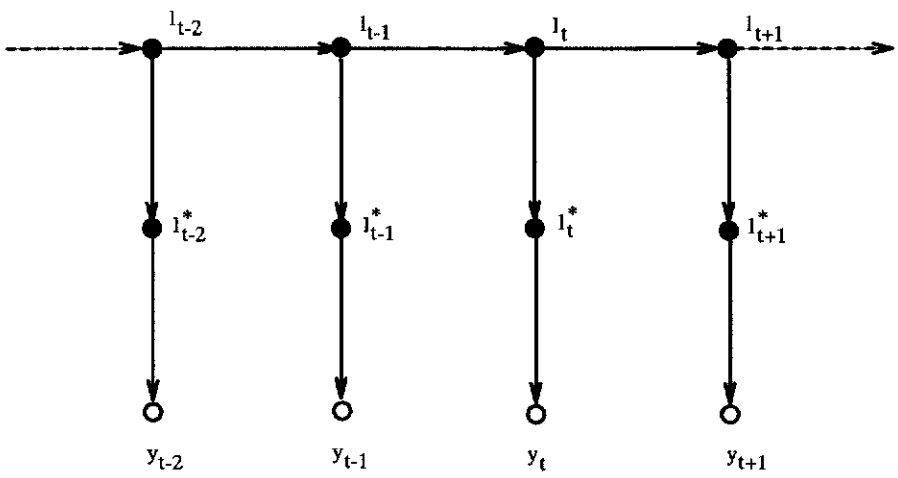
For MODEL3, the values of $l^*$ are unobserved and for MODEL4 the values of $l$ are unobserved: the EM algorithm is used to estimate the tables $p(l^*_{1990}|l^*_{1989})$, $p(l^*_{1991}|l^*_{1990})$, $p(l^*_{1992}|l^*_{1991})$ for MODEL3, and the tables $p(l_{1990}|l_{1989})$, $p(l_{1991}|l_{1990})$, $p(l_{1992}|l_{1991})$ for MODEL4. If sufficient ground validation sites were available, then this would have provided an alternative means of estimating the tables.

Fifty iterations of the EM algorithm were specified. Every fourth pixel of every fourth line was used in the EM operation. This sub-sampling scheme was adopted to reduce computation time, and results in only a sixteenth of the data being processed during the EM operation. The choice of every fourth line and pixel was made on the basis of computation time and the assumption that 48750 observations should be adequate to estimate the tables which have 121 parameters.

Having estimated the necessary unobserved parameters, the whole data were processed to form, for each model, land use/condition classifications for the years 1989, 1990, 1991 and 1992.




(a) MODEL3




(b) MODEL4

Figure 6.9: Temporal Models.

### 6.4.2 Spatial Models

In section 6.3, it was observed that *spatial smoothing* of the resulting classification occurred when remotely sensed data were combined with soil data. This smoothing was enhanced when the model parameters were estimated using the EM algorithm. In MODEL1 and MODEL2, the ground data partitioned the region into a number of relatively large (as compared to the pixel size of Landsat TM) homogeneous regions which provided the source of the smoothing. For this reason, this form of smoothing will be called *broad-scale* smoothing.

The amount of smoothing depends on the relative strength of the prior and the information derived from the spectral component of the model. In this way, it is not surprising that MODEL2–EM provided the most smoothing: the inclusion of the error estimates reduces the strength of the information derived from the spectral component of the model, and the EM algorithm iteratively favours the most common landcover encounterered/expected for each ground label.

While the combination of different data may improve the classifcation accuracy and spatial coherency of the resulting interpretation, errors (for example an isolated pixel labelled as lupin in the middle of a paddock sown to wheat) still occur. It is therefore of interest to include more explicitly the spatial correlation of neighbouring pixels.

#### Existing Spatial Schemes

Besag (1986) considered an iterative algorithm for including neighbourhood spatial context. For iteration $i, i = 1, \ldots, n$, the labelling of a pixel is influenced by the labels of the neighbouring pixels obtained from the previous iteration, that is iteration $i-1$. If $n(j)$ represents the neighbours of pixel $j$, and $|n_k^{i-1}(j)|$ represents the number of neighbours of pixel $j$ having label $l_k$ at iteration $i-1$, then the (prior) probability of class $k$ for iteration $i$ is given by:

$$p^i(l_k|n(j)) \propto \exp\{\alpha + \beta|n_k^{i-1}(j)|\}, \tag{6.5}$$

where $\alpha$ and $\beta$ are constants which control the amount of influence which the neighbourhood operations have in the classification process. That is, the probability of a pixel having label $l_k$ is proportional to the number of neighbouring pixels having the same label. This approach, along with other variations (refer to the original paper for more details), was considered in the context of remote sensing by Kiiveri and Campbell (1992), where tests on $50 \times 25$ pixels of artificial data were performed. Using the neighbour-prior approach, the authors report a classification accuracy of 94.5% as compared with 80.7% for the MLC, a significant improvement. The model proposed by Besag (1986) will be referred to as model MLC-NBR in the following, where NBR is short for neighbourhood.

Kiiveri and Campbell (1992) mention that the success of such methods rely on a good initial classification. MODEL1–EM and MODEL2–EM improved the salinity mapping accuracy, and hence could form the basis of an improved starting point, as compared with the MLC, for this purpose. Another benefit of these models is the ability of the broad-scale effect to relocate regions of incorrectly labelled pixels, which if used as a starting point for the MLC-NBR, would be likely to produce a false maximum ("clumps" of pixels tend to reinforce each other).

However, MODEL1–EM performed no better than the MLC for paddock-use mapping and MODEL2–EM actually significantly degraded the performance of mapping paddock-use and therefore is likely to be an inappropiate starting point for this goal. Other Bayesian models may be more appropriate.

To consider neighbourhood operations in the context of Bayesian networks, a graphical notation for neighbourhood operations is given in figure 6.10, which depicts the MLC-NBR. Here the variable associated with the pixel neighbourhood is shown as a lattice, to distinguish it from other variables.

### A New Generalised Scheme

A generalised means of incorporating neighbourhood operations into the classification process is considered, with the emphasis on models which combine multiple

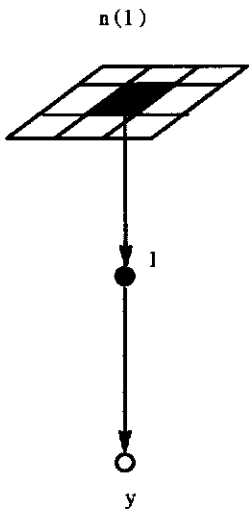



Figure 6.10: A graphical depiction of the neighbourhood model proposed by Besag (1986).

sources of data. The scheme is generalised in the sense that the model proposed by Besag (1986) is a special case. As an example, neighbourhood operations will be incorporated into MODEL3 and MODEL4, as both models are expected to improve classification accuracies for both paddock use and salinity mapping. However, the method employed to incorporate the neighbourhood operations could be applied to general forms of Bayesian networks.

The scheme introduced follows that proposed by Besag (1986), in the sense that it is iterative in nature and uses neighbouring pixel labels to influence the labeling of the current pixel.

Firstly a general Bayesian model, $G = (V, E)$ having parameterisation $\psi_{original}$, is assumed, which is the model used for combining data/evidence in the usual way. That is, if evidence is represented by $\psi_e$, then the evidence is combined as $\psi_{original} \otimes \psi_e$.

Next assume a set $v \subset V$ of $m$ variables $v = \{v_1, v_2, \ldots, v_m\}$ for which neighbourhood operations are desired. For each element of $v$, define a (probabilistic) mapping $\psi_{nbr}(v_j|n(v_j)); j = 1 \ldots m$ of a label of a pixel given the labelling of its neighbours. For example, such a mapping can be generated from equation 6.5, that is: $\psi_{nbr}(v_j = l_k \mid n(v_j)) \propto \exp(\alpha + \beta|n_k(v_j)|)$. The desired graph

is $G_{nbr} = (V \cup n(v_j), E \cup (n(v_j), v_j))$; $j = 1 \ldots m$. The classification proceeds as follows:

1. In the first iteration ($i = 0$), combine (process) the data in the usual way using $G$. That is, process all the data to form initial interpretations from which $\psi_{nbr}^{i+1}(v_j|n^i(v_j))$; $j = 1 \ldots m$ may be obtained. In the case of the model proposed by Besag (1986) and later considered by Kiiveri and Campbell (1992), the first iteration corresponds to performing a MLC, from which $\psi_{nbr}^{i+1}(v_j|n^i(v_j))$; $j = 1 \ldots m$ is obtained from equation 6.5.

2. For the next iteration, use $G_{nbr}$ to calculate $\psi_{original} \otimes \psi_e \otimes \psi_{nbr}^{i}(v|n^{i-1}(v))$. That is, as well as combining the data in the usual way, the information derived from the labels of neighbouring pixels is also combined.

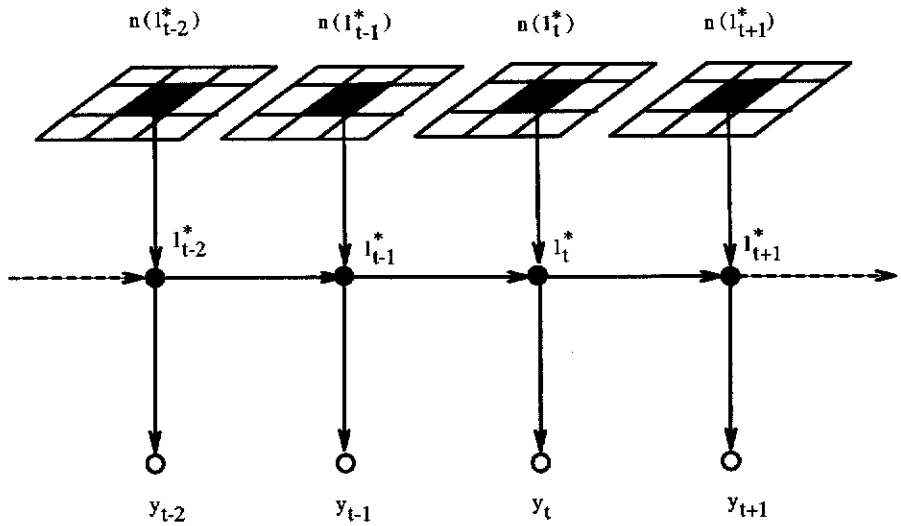
3. Repeat the above step for the desired number of iterations. Kiiveri and Campbell (1992) noted that most change occurred in the first few iterations of the neighbourhood operation.

To illustrate some aspects of the above approach, neighbourhood operations were incorporated into MODEL3 and MODEL4 to form MODEL5 and MODEL6 repsectively. The new models are depicted graphically in figure 6.11.

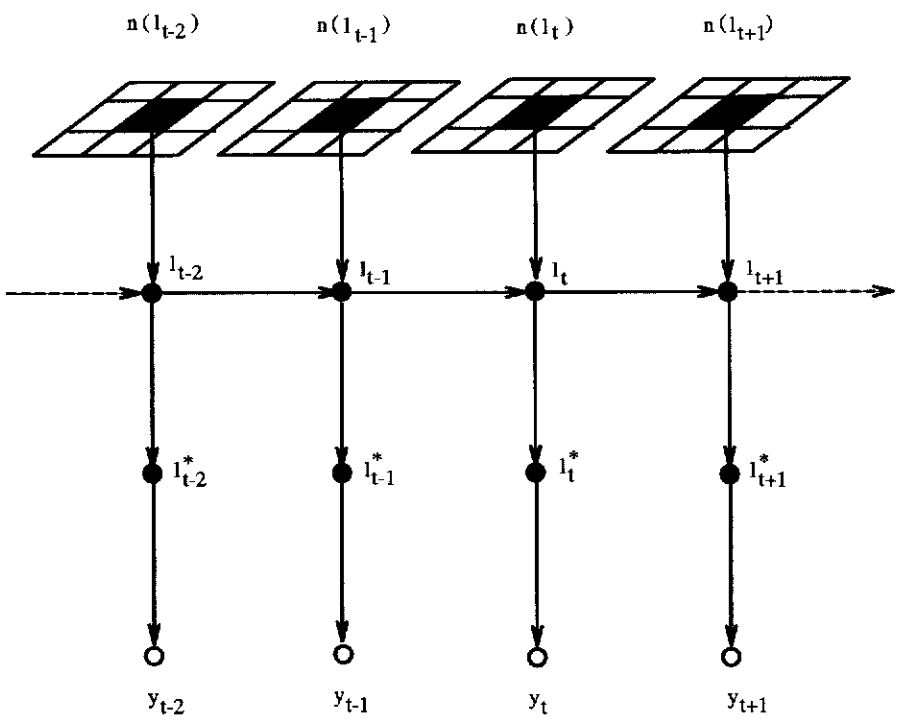
As observed from figure 6.11, neighbourhood operations are used to influence the labelling of pixels in each year. Thus in the first iteration, the data are combined using temporal rules, and then in subsequent operations both temporal and spatial rules are used.

Some comments on the above are required:

1. If $n(v_j) = \{v_{j1}, v_{j2}, \ldots, v_{j8}\}$ represents the eight neighbouring pixels of $v_j$, then an obvious choice for specifying $p(v_j|n(v_j))$ is $p(v_j|v_{j1}, v_{j2}, \ldots, v_{j8})$. However, this may become unreasonably large if too many classes are specified (for instance, if ten classes are specified, then the table would have size $10^5$). Therefore, some other specification, such as equation 6.5, is used.




(a) MODEL5




(b) MODEL6

Figure 6.11: Temporal / Spatial Models.

2. As described for the general case, an extra independence assumption is made than that which would normally be inferred from the graphical notation. Taking MODEL3 as an example, from the graphical notation one would expect the conditional distribution $p(l_{1990}|l_{1989}, n(l_{1990}))$ to be specified. However, for simplicity this distribution is formed as

$$p(l_{1990}|l_{1989}, n(l_{1990})) \propto p(l_{1990}|l_{1989})p(l_{1990}|n(l_{1990})).$$

The only justification for this is that it avoids the problem of specifying/estimating $p(l_{1990}|l_{1989}, n(l_{1990}))$ by specifying the simpler distributions for $p(l_{1990}|l_{1989})$ and $p(l_{1990}|n(l_{1990}))$ separately. In the general case, this assumption becomes more attractive, as it becomes less clear how to specify/estimate $p(v|pa(v)\backslash n(v), n(v))$.

### 6.4.3 Classifier Comparisons

#### Salinity Mapping Accuracy

The classifications were compared with the salinity ground truth; the classification accuracies are summarised in table 6.14.

MODEL3 had an overall error rate of 45%, as compared with the MLC (table 6.7) which had an error rate of 43%. MODEL4 performed better by this criterion, having an error rate of 38%. Of interest is the consistency of the area estimated to be *SaltSure*, as this estimate should not vary significantly for the time period considered and hence provides an indication of the performance of each model. From table 6.14, it is observed that the temporal consistency of the *SaltSure* class has been improved by MODEL4. For example, the MLC estimates of *SaltSure* over the period are roughly in the range 31000 to 64000 pixels, whereas the MODEL4 estimates are in the range 30000 to 49000 pixels. The range of values for MODEL3 are roughly the same (34000 to 67000) as for the MLC, although shifted upwards. This result parallels those obtained from comparing

| Salt Class | Model / Classifier | | | | | |
|---|---|---|---|---|---|---|
| | MODEL3 | MODEL4 | MODEL5 | MODEL6 | MODEL6* | MLC-NBR |
| | Year 1989 | | | | | |
| $\|SaltSure\|$ | 45387 | 42656 | 49856 | 44427 | 34035 | 50965 |
| $\|SaltSure_{err}\|$ | 20424 | 17489 | 21438 | 15994 | 9863 | 23256 |
| % Incorrect | 45 | 41 | 43 | 36 | 29 | 46 |
| $\|SaltMarginal\|$ | 24916 | 24103 | 19071 | 15612 | 17504 | 20276 |
| $\|SaltMarginal_{err}\|$ | 16445 | 14944 | 12396 | 9055 | 10348 | 12928 |
| % Incorrect | 66 | 62 | 65 | 58 | 59 | 64 |
| | Year 1990 | | | | | |
| $\|SaltSure\|$ | 67167 | 49095 | 69514 | 48038 | 37814 | 68595 |
| $\|SaltSure_{err}\|$ | 32911 | 17674 | 32672 | 15372 | 10716 | 35111 |
| % Incorrect | 61 | 45 | 47 | 32 | 28 | 51 |
| $\|SaltMarginal\|$ | 14658 | 8166 | 11461 | 5366 | 5106 | 16234 |
| $\|SaltMarginal_{err}\|$ | 8941 | 3675 | 6533 | 1663 | 1558 | 9207 |
| % Incorrect | 61 | 45 | 57 | 31 | 31 | 57 |
| | Year 1991 | | | | | |
| $\|SaltSure\|$ | 34493 | 29981 | 37277 | 33487 | 38891 | 31625 |
| $\|SaltSure_{err}\|$ | 11728 | 9294 | 11929 | 9711 | 10880 | 11138 |
| % Incorrect | 34 | 31 | 32 | 29 | 28 | 35 |
| $\|SaltMarginal\|$ | 13507 | 13092 | 11739 | 10663 | 3202 | 14489 |
| $\|SaltMarginal_{err}\|$ | 2836 | 2880 | 2113 | 1919 | 547 | 3309 |
| % Incorrect | 21 | 22 | 18 | 18 | 17 | 23 |
| | Year 1992 | | | | | |
| $\|SaltSure\|$ | 36742 | 36181 | 37277 | 37929 | 41197 | 36402 |
| $\|SaltSure_{err}\|$ | 10288 | 9769 | 11928 | 9482 | 11182 | 9738 |
| % Incorrect | 28 | 27 | 32 | 25 | 27 | 27 |
| $\|SaltMarginal\|$ | 19588 | 11135 | 11739 | 7963 | 5317 | 20427 |
| $\|SaltMarginal_{err}\|$ | 11361 | 5122 | 6809 | 3583 | 452 | 11630 |
| % Incorrect | 58 | 46 | 58 | 45 | 9 | 57 |
| | Total Errors, All years | | | | | |
| $\|SaltSure\|$ | 183789 | 157913 | 193924 | 163881 | 151937 | 187587 |
| $\|SaltSure_{err}\|$ | 75351 | 54226 | 77967 | 50559 | 42641 | 79243 |
| % Incorrect | 41 | 34 | 40 | 31 | 28 | 42 |
| $\|SaltMarginal\|$ | 72669 | 56496 | 54010 | 39604 | 31129 | 71426 |
| $\|SaltMarginal_{err}\|$ | 39583 | 26621 | 27851 | 16220 | 12905 | 37074 |
| % Incorrect | 54 | 47 | 40 | 31 | 41 | 52 |
| | Total Errors, All years/All Classes | | | | | |
| $\|SaltAffected\|$ | 256458 | 214409 | 247934 | 203485 | 183066 | 259013 |
| $\|SaltAffected_{err}\|$ | **114934** | **80847** | **105818** | **66779** | **55546** | **116317** |
| **% Incorrect** | **45** | **38** | **43** | **33** | **30** | **45** |

Table 6.14: Accuracy for salinity classification.

MODEL1 with MODEL2; the incorporation of the classification error rates allows more opportunity for the non-spectral component of the model to have an effect.

Comparing the result obtained from the MLC (table 6.7) with that of MODEL4 (table 6.14), it is observed that the total area identified as saline, ($|SaltAffected|$), by MODEL4 (203485 pixels) is approximately 21% less than that mapped by MLC (258604). The examples of the different classifications given in figure 6.12 help demonstrate the reason for this: the MLC misclassifies poor pastures and crops as salt-affected regions, whereas for MODEL4, the temporal rules largely reduced these errors, relabelling large misclassified areas of paddocks correctly.

Comparing the results (tables 6.7 and 6.14) of the MLC with MLC-NBR, MODEL3 with MODEL5, and MODEL4 with MODEL6, it is observed that the models with the neighbourhood operations outperform their non-neighbourhood counterparts, although the average improvements are rather modest: the average improvement for each comparison is in the vicinity of 2% to 5% less errors made. From the tables it is observed that the neighbourhood operations had a larger effect on the *SaltMarginal* class than on the *SaltSure* class. For example, in the period 1989 to 1992, 47% of pixels mapped as *SaltMarginal* by MODEL4 were in error, as opposed to MODEL6 having an error rate of 31%; a significant improvement. On the other hand, MODEL4 and MODEL6 produced error rates of 34% and 31% respectively for the *SaltSure* class over the same period. This result can be explained as follows: the temporal relationships concerning the *SaltSure* class are strongly defined (a high probability of *SaltSure* exists if the previous year was also *SaltSure*), whereas the rules governing the *SaltMarginal* class are not so strongly defined[2]. This, combined with the poorer spectral separation of *SaltMarginal* sites from other classes, results in many isolated pixels being incorrectly labelled as *SaltMarginal*. The neighbourhood operations help in this regard by reducing the number of these pixels.

[2]Marginally salt-affected areas are less easily characterised in the temporal sense as they, like non-affected areas, support crops and pastures.

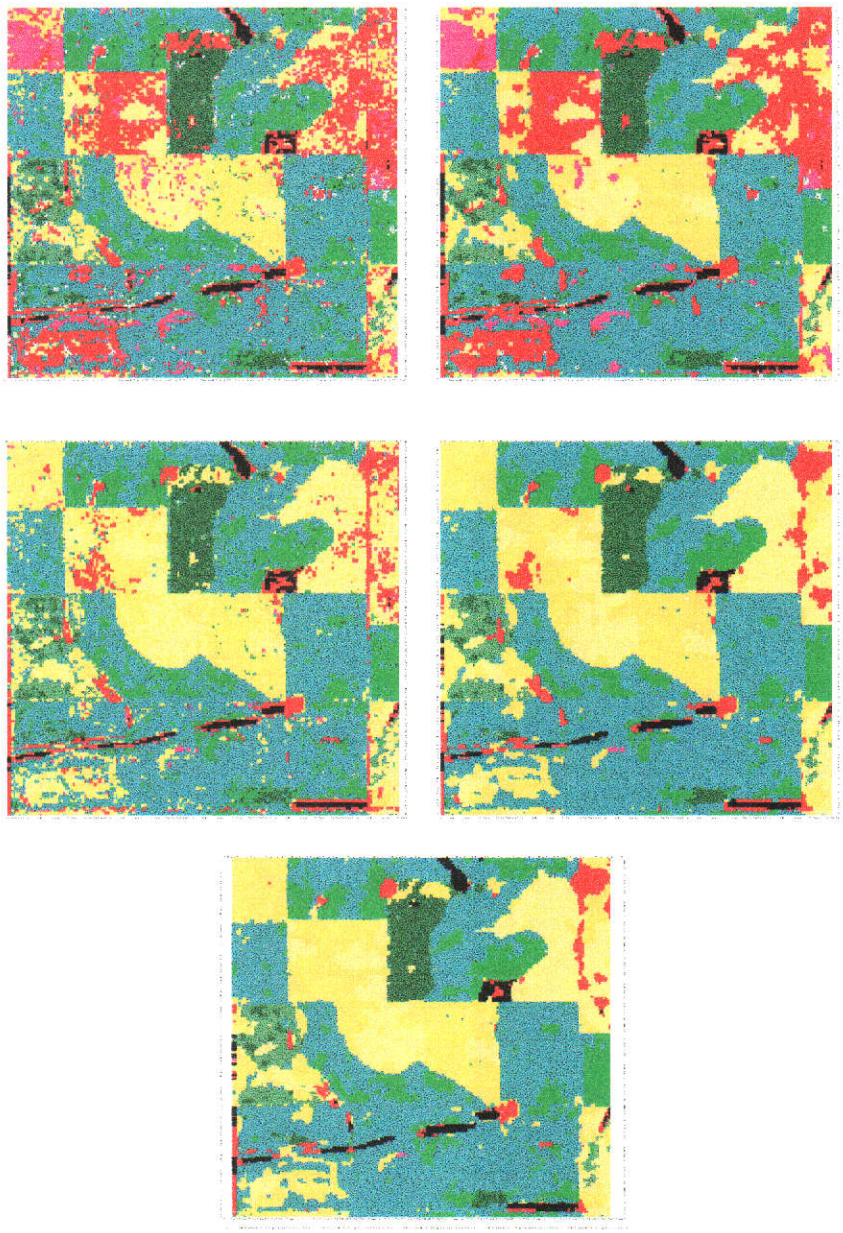


Figure 6.12: Model Comparison (1990): (top-left) MLC, (top-right) MLC-NBR, (middle left) MODEL4, (middle right) MODEL6 and (bottom) MODEL6*.

The effect of the neighbourhood operations is illustrated in figure 6.12. Comparing the MLC with MLC-NBR and MODEL4 with MODEL6 it is evident that the neighbourhood operations remove potentially spurious pixel labels and form regions that are more spatially coherent. Also of interest is the interaction of the temporal and spatial components of the models. For instance, it is observed that MLC produces large areas of misclassified pixels; application of neighbourhood operations (MLC-NBR) reinforces these errors. However, the temporal-spatial model largely avoids this problem: the temporal relationships correctly label these regions (broad-scale smoothing) before the neighbourhood operations locally reinforce spatial coherence.

Still of concern in temporal models is the relative weakness in the temporal coherency for the $SaltSure$ class for the years 1990 and 1991. From the MLC, it is observed that the $SaltSure$ class is grossly overestimated (spectrally) in 1990. Although the temporal models lessen this effect, it is still present. Examination of the estimated parameters for the probability of $SaltSure$ in the next year given that the label is $SaltSure$ in the current year gives the probabilities (MODEL4 and MODEL6) $p(SaltSure_{1990}|SaltSure_{1989}) = 0.92$, $p(SaltSure_{1991}|SaltSure_{1990}) = 0.53$ and $p(SaltSure_{1992}|SaltSure_{1991}) = 0.83$ which helps explain the lack of temporal consistency. That is, the overestimate of $SaltSure$ in 1990 meant that in the following year the area may or may not have been salt-affected, resulting in the low probability ($p(SaltSure_{1991}|\ SaltSure_{1990})$=0.53) which limits the temporal coherency of the model. This hypothesis is examined below.

On the assumption that the values of $l$ in MODEL4 and MODEL6 represent the *true* land cover class, the estimate $p(SaltSure_{1991}|SaltSure_{1990}) = 0.53$ appears to be too low. Given that recovery of severely salt-affected land is typically not achieved within one year, one would expect the figure $p(SaltSure_t|SaltSure_{t-1})$ to be closer to 1.0. The poor parameter estimates are the result of ignoring this prior knowledge. The parameters were re-estimated using this knowledge. Specifically, the parameters $p(SaltSure_t|SaltSuret-1)$, $t = 1990 \ldots 1992$, were

(heuristically) specified to be 0.98 before commencement of the EM algorithm. The EM algorithm was then applied (in the usual way), with the values of the parameters above forming initial estimates. The idea here is that although the parameters were subjectively defined, they are *refined* using all other available evidence. To gauge the effect of this on the final classification, results for MODEL6 using these new estimates were collected. In the results, MODEL6 with the new estimates will be referred to as MODEL6*.

From table 6.14, we observe that the temporal consistency of the *SaltSure* estimates has improved, with yearly estimates obtained from MODEL6* being in the range [34035,41197] as compared with the range [33487,48038] obtained from MODEL6. Further, unlike any of the previous models, the results trend in the right direction, showing a slight increase in the area of severely-affected land as time progresses. This is what would be expected for this region, as little remedial action had been taken. From the figures obtained in this section, the extent of severely salt-affected land in the Xantippe catchment increases each year by roughly 10% of what it was in the previous year.

Comparing the results (table 6.14) of MODEL6 with MODEL6*, we observe that overall, MODEL6* labelled 16% less pixels as *SaltSure* (42641 pixels) than did MODEL6 (50559 pixels) for the period 1989 to 1992.

### Paddock-Use Mapping Accuracy

The classifications were compared with the paddock-use ground truth; the results are summarised in tables 6.15, 6.16, 6.17 and 6.18, while table 6.19 summarises the overall performance of the models.

From tables 6.8, 6.9, 6.10 and 6.11, we observe that the MLC has an accuracy of 60%, 73%, 71% and 85% for the years 1989, 1990, 1991 and 1992 respectively. Comparing these accuracies with those summarised in tables 6.15, 6.16, 6.17, 6.18, and 6.19, it is observed that MODEL3, MODEL4, MODEL5 and MODEL6 consistently outperform the MLC. Comparing the results of the temporal models and

their corresponding neighbourhood versions, that is comparing the results for MLC with MLC-NBR, MODEL3 with MODEL5, and MODEL4 with MODEL6, it is observed that the models with neighbourhood operations consistently outperform those without. Table 6.19 summarises the average performance of the models, from which it is observed that the temporal rules increase the overall average accuracy by about 2–3% and the neighbourhood operations increase this further by another 3–4%, increasing the overall accuracy by about 6%.

While this gain is rather modest, the gain was consistently obtained, and the temporal rules increased the accuracy for the worst year (for the MLC) from 60% to 70%. Further, little effort was expended in obtaining the increases, amounting to performing the computations for the non-spectral part of the model approximately three times for MODEL3 and MODEL4 (to estimate the parameters via EM) and then for MODEL5 and MODEL6 another three times for the neighbourhood operations[3].

Finally, we observe that MODEL6* has the best overall average, identifying paddock-use with an accuracy of 79%.

### Spatial Uniformity

The spatial uniformity of the resulting interpretations was compared with that of the MLC; the results are presented in table 6.20. The uniformity measure adopted is the total number of adjacent pixels which have different classification labels. The values in the table represent the improvement in the uniformity measure for the new models as compared with the MLC, and results being expressed as percentages.

From the table, it is observed that the models which explicitly include neighbourhood operations, that is MLC–NBR, MODEL5 and MODEL6, significantly improve the spatial uniformity of the resulting interpretation, improving on the

[3]Note that potentially expensive calculations regarding the spectral part of the model need only be performed once.

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | % Correct |
| | MODEL3 | | | | **Average 62** |
| Wheat | 5791 | 1 | 2754 | 1667 | 57 |
| Lupins | 32 | 155 | 215 | 38 | 35 |
| Pasture | 402 | 139 | 3975 | 877 | 74 |
| | MODEL4 | | | | **Average 65** |
| Wheat | 6526 | 604 | 1565 | 1518 | 64 |
| Lupins | 4 | 393 | 5 | 38 | 89 |
| Pasture | 527 | 634 | 3519 | 713 | 65 |
| | MODEL5 | | | | **Average 67** |
| Wheat | 6203 | 0 | 2681 | 1329 | 61 |
| Lupins | 8 | 167 | 240 | 25 | 38 |
| Pasture | 333 | 133 | 4358 | 569 | 81 |
| | MODEL6 | | | | **Average 70** |
| Wheat | 7327 | 584 | 1193 | 1109 | 72 |
| Lupins | 0 | 437 | 0 | 3 | 99 |
| Pasture | 511 | 656 | 3833 | 393 | 71 |
| | MODEL6* | | | | **Average 72** |
| Wheat | 7215 | 601 | 1217 | 1180 | 71 |
| Lupins | 0 | 424 | 0 | 6 | 99 |
| Pasture | 488 | 660 | 3869 | 376 | 72 |
| | MLC-NBR | | | | **Average 63** |
| Wheat | 5751 | 0 | 3021 | 1441 | 56 |
| Lupins | 11 | 154 | 249 | 26 | 35 |
| Pasture | 303 | 118 | 4234 | 738 | 79 |

Table 6.15: Landcover Accuracies, 1989.

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | % Correct |
| | MODEL3 | | | | **Average 74** |
| Wheat | 4255 | 6 | 301 | 469 | 85 |
| Lupins | 91 | 1736 | 618 | 144 | 67 |
| Pasture | 375 | 82 | 5832 | 1805 | 72 |
| | MODEL4 | | | | **Average 79** |
| Wheat | 4322 | 5 | 467 | 237 | 86 |
| Lupins | 152 | 1637 | 769 | 31 | 63 |
| Pasture | 535 | 104 | 6750 | 705 | 83 |
| | MODEL5 | | | | **Average 76** |
| Wheat | 4376 | 3 | 261 | 391 | 87 |
| Lupins | 57 | 1763 | 628 | 141 | 68 |
| Pasture | 256 | 54 | 6010 | 1774 | 74 |
| | MODEL6 | | | | **Average 82** |
| Wheat | 4389 | 2 | 457 | 183 | 87 |
| Lupins | 74 | 1737 | 767 | 11 | 67 |
| Pasture | 320 | 71 | 7032 | 671 | 87 |
| | MODEL6* | | | | **Average 84** |
| Wheat | 4405 | 2 | 520 | 104 | 88 |
| Lupins | 75 | 1725 | 788 | 1 | 67 |
| Pasture | 312 | 63 | 7059 | 660 | 87 |
| | MLC-NBR | | | | **Average 76** |
| Wheat | 4281 | 3 | 243 | 504 | 85 |
| Lupins | 54 | 1780 | 581 | 174 | 69 |
| Pasture | 202 | 55 | 5842 | 1995 | 72 |

Table 6.16: Landcover Accuracies, 1990.

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | % Correct |
| | MODEL3 | | | | **Average 74** |
| Wheat | 6023 | 1 | 201 | 67 | 96 |
| Lupins | 42 | 1079 | 250 | 1 | 79 |
| Pasture | 2967 | 54 | 4789 | 572 | 57 |
| | MODEL4 | | | | **Average 70** |
| Wheat | 5797 | 69 | 396 | 30 | 92 |
| Lupins | 37 | 781 | 553 | 1 | 57 |
| Pasture | 3229 | 166 | 4632 | 355 | 55 |
| | MODEL5 | | | | **Average 77** |
| Wheat | 6236 | 0 | 41 | 15 | 99 |
| Lupins | 1 | 1180 | 191 | 0 | 86 |
| Pasture | 3004 | 6 | 4941 | 431 | 59 |
| | MODEL6 | | | | **Average 72** |
| Wheat | 6026 | 84 | 169 | 13 | 96 |
| Lupins | 0 | 808 | 564 | 0 | 59 |
| Pasture | 3277 | 85 | 4647 | 373 | 55 |
| | MODEL6* | | | | **Average 71** |
| Wheat | 5921 | 86 | 212 | 3 | 94 |
| Lupins | 0 | 821 | 551 | 0 | 60 |
| Pasture | 3244 | 83 | 4690 | 365 | 56 |
| | MLC-NBR | | | | **Average 77** |
| Wheat | 6088 | 0 | 173 | 31 | 97 |
| Lupins | 7 | 1289 | 76 | 0 | 94 |
| Pasture | 2774 | 94 | 5002 | 512 | 60 |

Table 6.17: Landcover Accuracies, 1991.

| True Class | Mapped Class | | | | |
|---|---|---|---|---|---|
| | Wheat | Lupins | Pasture | Other | % Correct |
| | MODEL3 | | | | **Average 87** |
| Wheat | 6824 | 56 | 355 | 159 | 92 |
| Lupins | 0 | 482 | 1 | 5 | 99 |
| Pasture | 439 | 64 | 6578 | 1083 | 81 |
| | MODEL4 | | | | **Average 86** |
| Wheat | 6989 | 85 | 205 | 115 | 95 |
| Lupins | 0 | 482 | 6 | 0 | 99 |
| Pasture | 1293 | 60 | 6307 | 504 | 77 |
| | MODEL5 | | | | **Average 91** |
| Wheat | 7121 | 42 | 112 | 119 | 96 |
| Lupins | 0 | 486 | 1 | 1 | 100 |
| Pasture | 311 | 62 | 6936 | 855 | 85 |
| | MODEL6 | | | | **Average 89** |
| Wheat | 7219 | 51 | 35 | 89 | 98 |
| Lupins | 0 | 485 | 3 | 0 | 99 |
| Pasture | 1026 | 59 | 6583 | 496 | 81 |
| | MODEL6* | | | | **Average 89** |
| Wheat | 7196 | 50 | 33 | 115 | 97 |
| Lupins | 0 | 485 | 3 | 0 | 99 |
| Pasture | 1033 | 59 | 6561 | 511 | 80 |
| | MLC-NBR | | | | **Average 90** |
| Wheat | 6988 | 48 | 236 | 122 | 95 |
| Lupins | 0 | 484 | 0 | 4 | 99 |
| Pasture | 272 | 63 | 6927 | 902 | 85 |

Table 6.18: Landcover Accuracies, 1992.

| MLC | MODEL3 | MODEL4 | MODEL5 | MODEL6 | MODEL6* | MLC-NBR |
|---|---|---|---|---|---|---|
| 72 | 74 | 75 | 78 | 78 | 79 | 76 |

Table 6.19: Average Paddock-Use Accuracies (%), 1989–1992.

| | MODEL3 | MODEL4 | MODEL5 | MODEL6 | MODEL6* | MLC-NBR |
|---|---|---|---|---|---|---|
| % Improvement, 1989 | 3 | 5 | 48 | 54 | 52 | 46 |
| % Improvement, 1990 | 8 | 13 | 45 | 53 | 50 | 40 |
| % Improvement, 1991 | 13 | 15 | 53 | 60 | 62 | 47 |
| % Improvement, 1992 | 9 | 21 | 46 | 61 | 63 | 41 |
| % Average | 9.7 | 13.5 | 48.0 | 57.0 | 56.8 | 43.5 |

Table 6.20: Improvement in Spatial Uniformity (%), 1989–1992.

MLC by more that 43% and their non-neighbourhood counterparts by some 40%.

Some spatial consistency results from the temporal rules, with MODEL3 and MODEL4 on average producing results which are 10% and 13% *smoother* (on the basis of the uniformity measure) than those obtained from the MLC. This effect is in part due to the temporal rules modifying the outcome for many isolated pixels which were incorrectly labelled by the MLC.

This is demonstrated in figure 6.12, which shows an example of the 1990 classification obtained from the MLC and from MODEL4. In this figure, the saline classes (represented by red and pink) are grossly overestimated by the MLC for this year, typically being confused with pasture (represented by yellow). Because these errors are not made for each year, the temporal rules have the effect of correctly relabelling many of these erroneous, isolated pixels.

## 6.5 A Knowledge-Based Approach to Combining Multiple-Sources of Data

In this section, a model which incorporates many diverse sources of information is conceived. A knowledge–based approach to constructing the Bayesian network is adopted, and will be described.

### 6.5.1 Knowledge Elicitation

Knowledge was gained from consultation with farmers, land care technicians and hydrologists, and identified the types of relationships that were likely to exist. As this thesis is not intended as an expose on farming practices and the process of salinity, the factors will be described more in a conceptual rather than a rigorous manner.

The following factors were identified as being important:

1. Land cover condition can be described as good or poor, with poor condition the result of factors including salinity, wind and water erosion, and farmer management practices such as heavy grazing of pasture. Poor land condition in this context is characterised by the absence of land cover, or in other words, bareness.

2. Apart from the cases where salt-tolerant vegetation such as samphire predominate, lack of vegetation is a feature of salinity, erosion and other classes of poor land condition. To be able to attribute poor land condition to the effects of salinity, we must be able to distinguish saline from non-saline causes.

3. Features such as overgrazing are the result of a single year's management practice, whereas salinity persists from year to year. Further, crop rotation will change land use yearly; in some years, a paddock may be cropped rather than grazed. Salinity will tend to have a more stable temporal pattern than grazing and other management effects.

4. Landform morphology is an important factor in the process of salinisation, eg. salinity is more likely to occur in valleys than on hilltops. Further, some parts of a valley, eg. points of convergent flow, are more prone to salinity than others.

5. Some regions of salinity are the result of a change in hydraulic conductivity on a flow path. For example, small salt scalds appear at soil boundaries.

6. Soils and landform are important factors in the condition of landcovers, eg. light sandy soils are more prone to wind erosion than heavy clay soils. This type of effect may be amplified by the position of the soil in the landscape.

Based upon the factors identified, data were obtained and processed to form variables useful as evidence.

### 6.5.2 Data Sets

The soil type image (figure 6.1) was used for two purposes. Firstly, an image of three broad categories of soil types was produced (figure 6.13) – DeepSands, ClayFlats and Others. These are used in determining relationships governing gross soil characteristics. Secondly, an image (figure 6.14) reflecting the uncertainty in position of a soil boundary was generated by distance-weighting the original soil boundary image. The following formula was used:

$$P(SoilChange = yes) = 0.5 - x/100 \text{ for } 0 \leq x \leq 100,\ 0 \text{ otherwise.}$$

where $x$ is the distance (m) from the original soil boundary. This reflects the fact that soil boundaries may in some cases be better represented as a transition zone rather than as a single line on a map.

The salinity map (figure 6.18) provided by the farmers was used as ground truth.

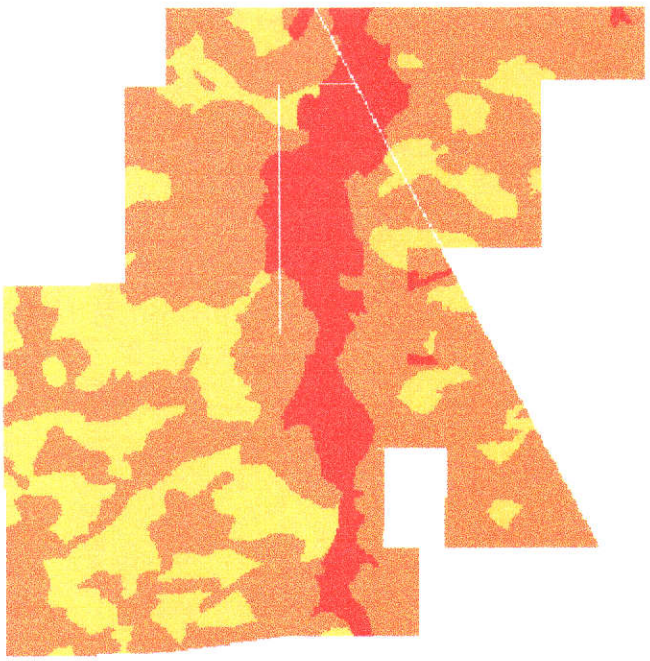


Figure 6.13: Soil information grouped into three classes.

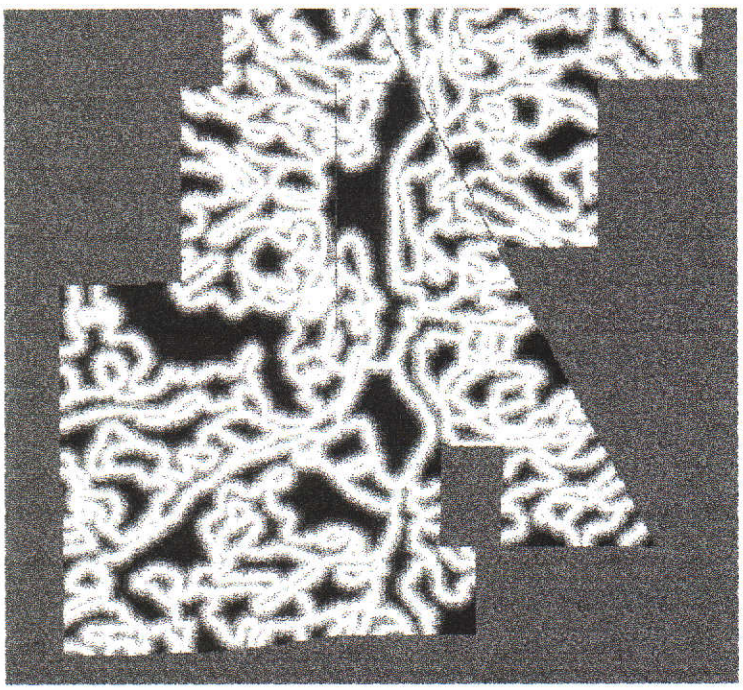


Figure 6.14: Probability of a soil change ($P(SoilChange)$), white = yes, black = no.

Figure 6.15: Probability of a convergent flow point ($P(ConvergentFlow)$), white = yes, black = no.

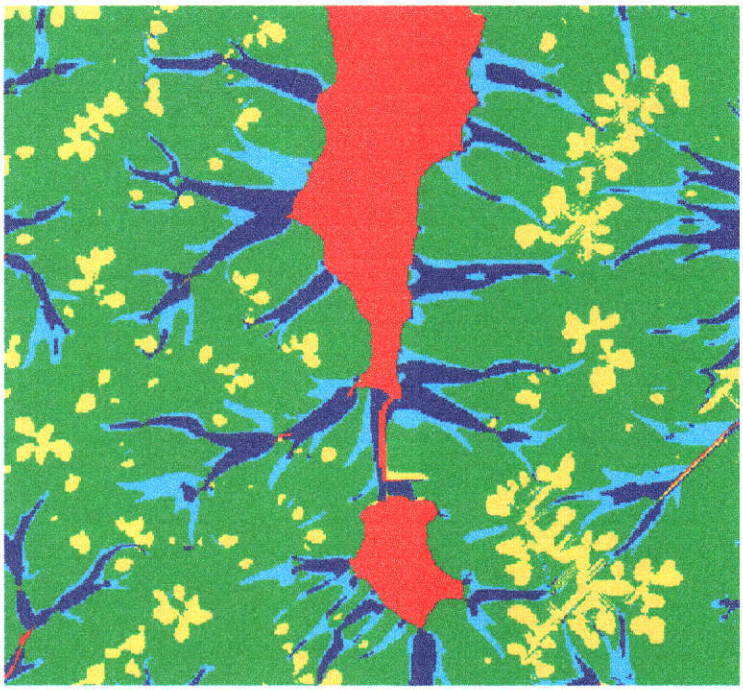


Figure 6.16: Stratified *Water Accumulation* image.

Figure 6.17: Flat Area Classification.

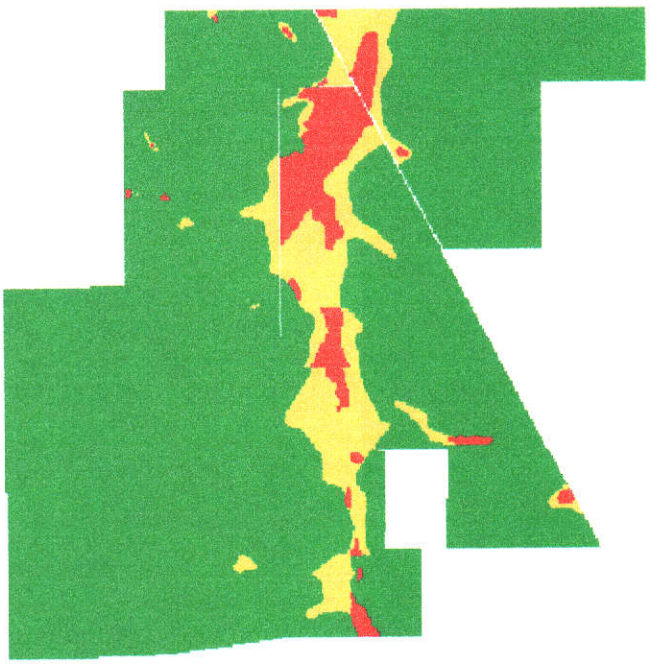


Figure 6.18: Salinity Ground Truth.

The ALLFLOW3 algorithm provided a landform partitioning, shown in figure 6.16. The valley systems were extracted from the landform partition image, and a probability image of convergent flow points was determined in the same manner as for the soil boundary image. The resulting image is given in figure 6.15.

### 6.5.3 Temporal Aspects of Land-Use

To understand what forms of temporal landuse relationships are likely to exist and what their effects on the results would be, farmer cropping records were obtained at selected sites within the MKG study area and their temporal patterns were examined. In all, paddock records for 16 farms were available. Typically, these records contained information on whether the paddock was planted to cereals (such as wheat, barley and lupins) or to some form of pasture. The condition of the landcover was usually not noted (similarly the variety of each cover), hence only the fact that a landcover was, say, lupins was recorded.

The data were analysed for temporal transitions between the major paddock cover types: wheat, lupins and pasture. Transitions between these landcovers are summarised in table 6.21.

From the table, it is observed that the strongest associations relate to the planting of lupins. For example, wheat is almost always planted after lupins (95% of the time) and lupins are never planted in two sucessive years. Similarly, lupins are typically planted after wheat (62% of the time) and pasture or wheat typically follow pasture.

Apart from the case of a landcover following lupins, more careful examination of the individual farm cropping rotations reveals a great range of different transition probabilities. Consider crop types following wheat on two farms, f9 and f3 say. For f9, lupins typically follow wheat (90% of the time) whereas for farm f3, lupins or pasture are almost equally likely to follow wheat. This is not surprising, considering that different farmers use different cropping rotations, and individual paddocks may be used for different purposes. Further, although not explicitly pre-

| Farm | Landuse Transitions | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | P-P | P-W | P-L | W-L | W-W | W-P | L-P | L-W | L-L |
| f1 | 15 | 16 | 0 | 13 | 15 | 23 | 0 | 16 | 0 |
| | 0.48 | 0.52 | 0.0 | 0.25 | 0.29 | 0.46 | 0.0 | 1.0 | 0.0 |
| f2 | 4 | 1 | 0 | 11 | 1 | 1 | 0 | 10 | 0 |
| | 0.8 | 0.2 | 0.0 | 0.84 | 0.08 | 0.08 | 0.0 | 1.0 | 0.0 |
| f3 | 64 | 23 | 1 | 24 | 3 | 23 | 1 | 30 | 0 |
| | 0.73 | 0.26 | 0.01 | 0.48 | 0.06 | 0.46 | 0.03 | 0.97 | 0.00 |
| f4 | 13 | 17 | 0 | 15 | 20 | 12 | 0 | 13 | 0 |
| | 0.43 | 0.57 | 0.00 | 0.32 | 0.42 | 0.26 | 0.00 | 1.0 | 0.0 |
| f5 | 13 | 36 | 5 | 35 | 12 | 2 | 0 | 8 | 0 |
| | 0.24 | 0.67 | 0.09 | 0.71 | 0.25 | 0.04 | 0.00 | 1.00 | 0.00 |
| f6 | 3 | 9 | 0 | 10 | 3 | 3 | 0 | 4 | 0 |
| | 0.25 | 0.75 | 0.00 | 0.62 | 0.19 | 0.19 | 0.00 | 1.00 | 0.00 |
| f7 | 2 | 6 | 0 | 5 | 0 | 5 | 0 | 6 | 0 |
| | 0.25 | 0.75 | 0.00 | 0.50 | 0.00 | 0.50 | 0.00 | 1.00 | 0.00 |
| f8 | 16 | 11 | 0 | 9 | 10 | 11 | 0 | 10 | 0 |
| | 0.59 | 0.41 | 0.00 | 0.30 | 0.33 | 0.37 | 0.00 | 1.00 | 0.00 |
| f9 | 17 | 89 | 2 | 82 | 8 | 1 | 1 | 2 | 0 |
| | 0.17 | 0.81 | 0.02 | 0.90 | 0.09 | 0.01 | 0.33 | 0.67 | 0.00 |
| f10 | 5 | 16 | 0 | 20 | 4 | 0 | 0 | 0 | 0 |
| | 0.24 | 0.76 | 0.00 | 0.83 | 0.17 | 0.00 | - | - | - |
| f11 | 3 | 12 | 0 | 10 | 8 | 1 | 0 | 2 | 0 |
| | 0.20 | 0.80 | 0.00 | 0.52 | 0.42 | 0.06 | 0.00 | 1.00 | 0.00 |
| f12 | 25 | 33 | 0 | 32 | 8 | 7 | 0 | 8 | 0 |
| | 0.42 | 0.57 | 0.01 | 0.61 | 0.28 | 0.11 | 0.00 | 1.00 | 0.00 |
| f13 | 45 | 22 | 0 | 26 | 6 | 1 | 0 | 2 | 0 |
| | 0.67 | 0.33 | 0.00 | 0.79 | 0.18 | 0.03 | 0.00 | 1.00 | 0.00 |
| f14 | 19 | 36 | 0 | 34 | 18 | 21 | 1 | 21 | 0 |
| | 0.35 | 0.65 | 0.00 | 0.46 | 0.25 | 0.29 | 0.05 | 0.95 | 0.00 |
| f15 | 1 | 21 | 0 | 15 | 2 | 10 | 5 | 4 | 0 |
| | 0.05 | 0.95 | 0.00 | 0.56 | 0.07 | 0.37 | 0.56 | 0.44 | 0.00 |
| f16 | 0 | 10 | 3 | 9 | 16 | 1 | 0 | 3 | 0 |
| | 0.00 | 0.77 | 0.23 | 0.35 | 0.61 | 0.04 | 0.00 | 1.00 | 0.00 |
| Total | 0.41 | 0.57 | 0.02 | 0.62 | 0.18 | 0.20 | 0.05 | 0.95 | 0.00 |

Table 6.21: Farm–paddock transition characteristics. For each farm the first of the two rows give the transition counts (in number of paddocks) and the second of the two rows the associated transition probabilities.

sented here, examination of the cropping records shows that cropping rotations (2, 3 and 4 years) are not strictly adhered to.

With these points in mind, temporal rules regarding cropping rotations need to be incorporated on a paddock-by-paddock basis. That is, each paddock should have its own set of transition probabilities. However, paddock boundaries were not available for the majority of the MKG region, or for the sub-catchment in question, and hence it appeared of limited value to incorporate temporal rules on specific landcovers[4].

For this reason, classes which would be expected to be more reliable (temporally) were devised for the incorporation of temporal rules. The classes (described in detail in the next section) chosen are *Normal*, *Saline* and *PoorProductivity*, which typically would be expected to be more uniform across farms than say classes closely related to crop rotations.

### 6.5.4 Network Configuration

Drawing on the knowledge elicited in sections 6.5.1 and 6.5.3, the network of figure 6.19 was conceived. In this figure, graph nodes have been drawn as tables, along with the node state labels and state probabilities. For convenience, the hollow circles representing the Landsat data have been dropped.

The principal aim of the network is to determine whether or not the land is affected by salinity for the time period considerred; this occurs at node *LandConditionStatus* (*LCS*). This node has the states *Normal*, *Saline*, and *PoorProductivity*. The *Normal* state represents land that has no signs of salinity and is in good condition. The *Saline* state represents land that is salt-affected, and the *PoorProductivity* state represents land that has consistently poor per-

[4]Recent work by Flack (1995) has considered some aspects of paddock segmentation of Landsat TM data in the MKG area. A catch-twenty-two exists here: on the one hand, we wish to improve mapping accuracies by incorporating paddock-based temporal rules, while on the other hand, segmentation relies on a fairly accurate classification to extract the paddocks to which we wish to apply the temporal rules.

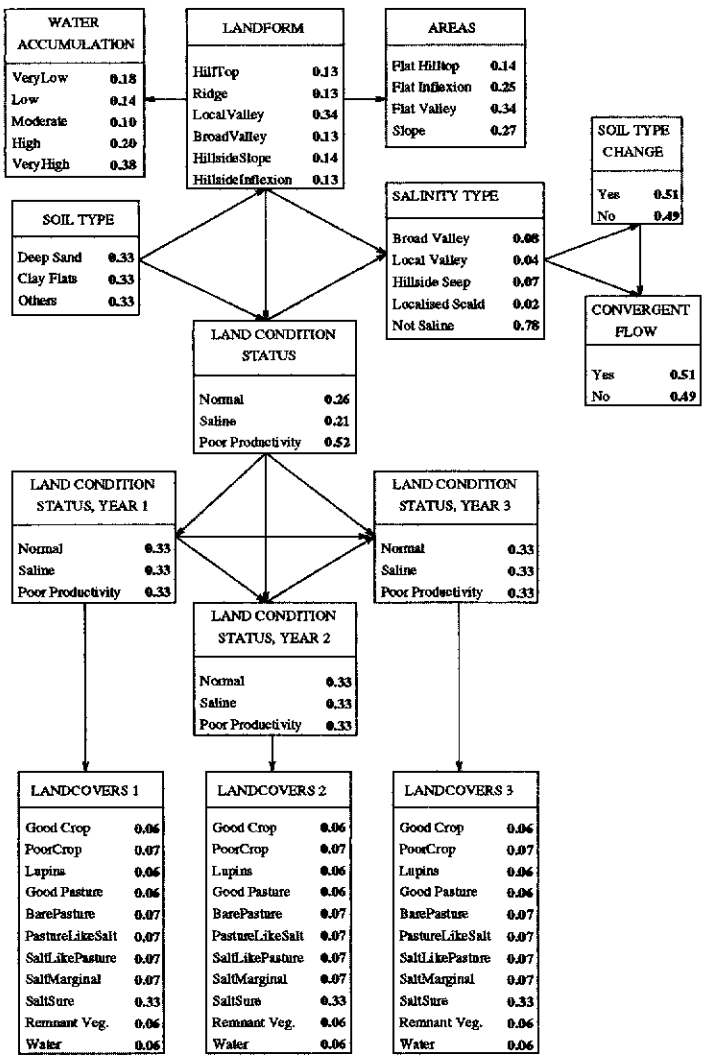



Figure 6.19: Graphical depiction of the network before training. Here variables are drawn as tables, and the variables' prior probabilities are typed in bold font.

formance not directly attributable to salinity.

The states of nodes *LandConditionStatus*, Year 1 (1989), 2 (1990) and 3 (1991) ($LCS1 - LCS3$) have the same definitions and represent the landcover condition for the three years. The manner in which these nodes are connected means that any permutations of the states of the nodes can be uniquely represented. That is, for each individual year, the network has an assessment of what state the land is in and uses combinations of these states to determine the true land state.

Nodes *LandCovers*1 to *LandCovers*3 ($LC1 - LC3$) have states that are the class labels as defined for MLC. In this sense, $LCS1$ represents $LC1$ grouped into broader classes.

Node *SoilType* ($S$), which has states *DeepSand*, *ClayFlats*, and *Others*, represents the soil types pooled into three classes. These soils, in combination with the landform categories defined in *LandForm* ($LF$), can be used to define rules governing $LCS$.

$LF$ represents the network's assessment of landform, and *SalinityType* ($ST$) uses this assessment along with the network's assessment of $LCS$ to characterise the forms of salinity, so that parameters affecting each different type of salinity can be incorporated into the model. Convergence of landform, *ConvergentFlow* ($CF$), and soil type transitions, *SoilTypeChange* ($STC$), are considered on the salinity subclasses.

Assessment of the true $LF$ is primarily influenced by *WaterAccumulation* ($WA$), *Areas* and $S$. In this way, $LF$ is not directly observed but its state is derived from what we observe at other nodes. For example, there is a very strong association between $S = ClayFlats$ and $LF = BroadValley$. Similarly, there is a strong relationship between $LF = Hilltop$ and $WA = VeryLow$. Known associations are a starting point for defining the parameters (strength of rules) of the network.

### 6.5.5 Parameter Determination

The network structure was conceived to represent a number of rules and associations presented by experts. Reasoning with uncertainty requires that estimates of rule strengths be obtained.

One approach is to have human experts give their intuitive feeling for what the rule strengths may be and then use these numbers directly. Where multiple experts are consulted, some means of pooling the results needs to be devised (see for example Ngwenyama and Bryson (1992)). This approach seems most appropriate where there are little ready data available for estimating the parameters directly.

A problem with the above approach is that it can easily lead to a network with probabilities that are intuitively in conflict with what would be expected (Osherson et al., 1994). This result reflects the fact that knowledge acquisition must be performed with care in such circumstances, and is a statement that modellers sometimes have difficulty deriving good probability estimates from experts[5].

Where much sample data are available, using the data to estimate the parameters is appropriate. When observed and unobserved nodes are present, then an EM algorithm may be used to estimate the necessary counts.

### 6.5.6 Application of the EM Algorithm – Training the Network

Knowledge of known associations was used where possible to form initial parameter estimates. When associations were not known, then a uniform distribution was specified for the rules in question, which means in effect that no rules were defined.

As an example of a rule, it is known that $ClayFlats$ essentially maps the bottom of a $BroadValley$, and as the soil maps were mapped with greater accuracy than the other $LF$ surrogates, this association should be given a high weighting. The rule used was $P(BroadValley|ClayFlats) = 0.99$, with the remaining weight

[5]Initial attempts at manually specifying the rule strengths showed this to be the case.

given to $P(LocalValley|ClayFlats) = 0.01$ and all other combinations of these variables given zero weight.

Rules were specified for the tables: $P(WA|LF)$, $P(AREAS|LF)$, $P(ST|LF, LCS)$, $P(LF|S)$, $P(LCS3|LCS, LCS1, LCS2)$, $P(LC1|LCS1)$, $P(LC2|LCS2)$ and $P(LC3|LCS3)$.

No rules were specified for the tables $P(LCS|LF, S)$, $P(S)$, $P(STC|ST)$ and $P(CF|ST, STC)$, as it was unclear to the experts what the values of these tables should be.

The input images shown in figures 6.13, 6.14, 6.15, 6.16 and 6.17 were mapped to the network nodes $S$, $STC$, $CF$, $WA$ and $AREAS$ respectively. Nodes $LC1$, $LC2$, and $LC3$ represent the landcovers obtained from classification of Landsat TM data. The spectral estimates used to interpret the Landsat TM data are the same as those used in the previous section.

To aid training, an extra node, $TruthSalinityMap$ ($TSM$), and a directed edge, $(LCS, TSM)$, were added to incorporate the truth image shown in figure 6.18. That is the truth image became an input during training. The image was grouped into two classes, $Saline$ and $Normal$, where the class $PotentiallySaline$ of figure 6.18 was reclassed as $Normal$, as this is the present state of the land. The extra link represented the rule $P(TSM|LCS)$ and had weights representing the following scenario: the areas mapped as $Saline$ in figure 6.18 are either likely to be $Saline$ or $PoorProductivity$ but not $Normal$ at $LCS$; the areas interpreted as $Normal$ in $TSM$ are likely to be $Normal$ or $PoorProductivity$ in $LCS$ but not $Saline$. In this way, the salinity information is used to separate the broad classes, $Normal$ and $Saline$, but recognises that $TSM$ makes no direct statement about $PoorProductivity$.

With the extra link in place, (heuristically) three iterations of the EM algorithm were performed on the top half of the image data and the updated tables were saved. The link added above was then removed (removing image 6.18 as input) and the data were processed using the network of figure 6.20, which incor-

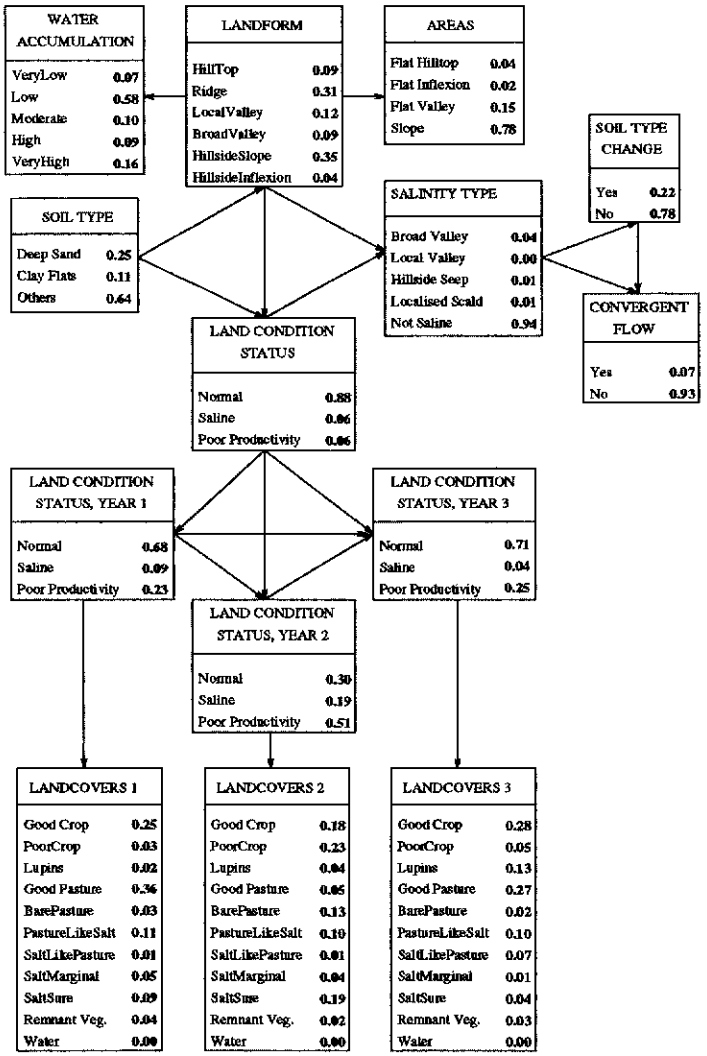



Figure 6.20: Graphical depiction of the network after training. Note the change (compare figure 6.19) in the variables' prior probabilities.

porates the updated tables.

### 6.5.7 Results

A summary of the accuracy of the map produced by the network as compared to the technician is given in tables 6.22, 6.23 and 6.24. The maps produced by the network are shown in figures 6.21 and 6.22, which show the most likely class label for $LCS$ and the probability for $LCS = Saline$ respectively.

As can be seen from comparing figure 6.21 with figure 6.18, there is good visual

| Network labels | Technicians' labels | | | Total |
|---|---|---|---|---|
| | NotAffected | PotAffected | SaltAffected | |
| NotAffected | 73569 | 7030 | 1956 | 82555 |
| Saline | 181 | 956 | 1901 | 3038 |
| Total | 73750 | 7986 | 3857 | 85593 |

Table 6.22: Network Mapping Accuracies: Network vs Technician.

| Network labels | Technicians labels | | |
|---|---|---|---|
| | NotAffected | PotAffected | SaltAffected |
| NotAffected | 0.89 | 0.09 | 0.02 |
| Saline | 0.06 | 0.31 | 0.63 |

Table 6.23: Network Mapping Accuracies: $P(Technician|Network)$.

| Network labels | Technicians labels | | |
|---|---|---|---|
| | NotAffected | PotAffected | SaltAffected |
| NotAffected | 0.997 | 0.88 | 0.51 |
| Saline | 0.003 | 0.12 | 0.49 |

Table 6.24: Network Mapping Accuracies: $P(Network|Technician)$.

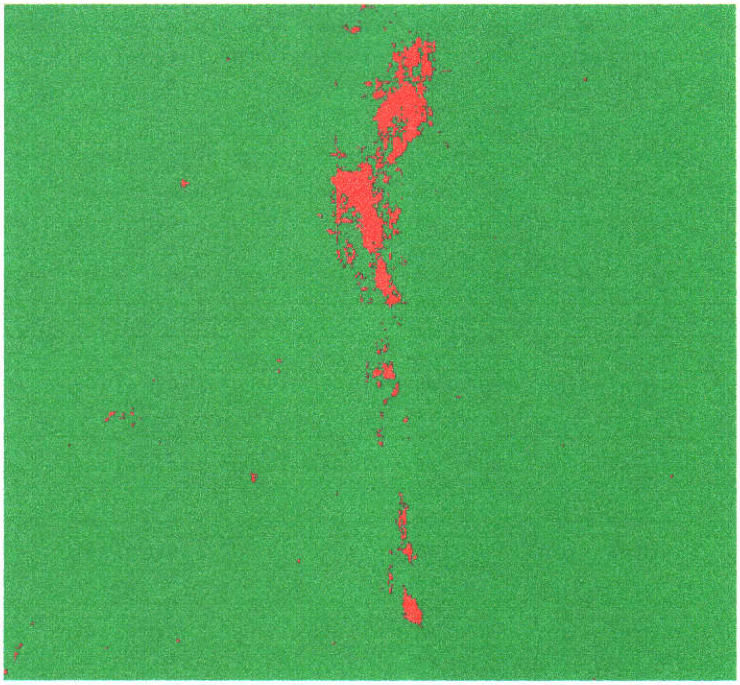


Figure 6.21: Network output: Salinity map. *Saline* areas are coloured red and *Normal* areas are coloured green. Note that *PoorProductivity* is never the most likely label.

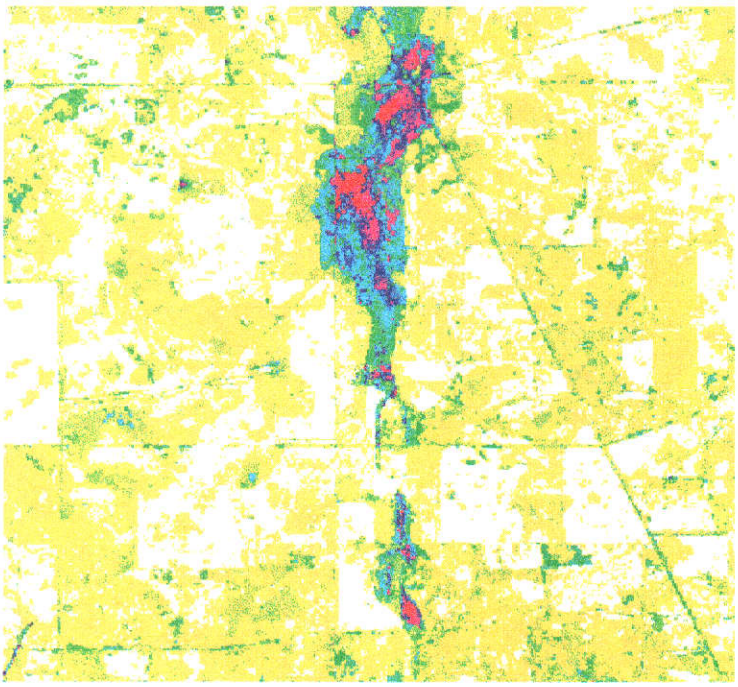


Figure 6.22: Network output: $P(Salinity)$. The map is coloured red, which indicates a high probability of salinity, through blue, cyan, green, yellow and then white which indicates a low probability of salinity.

agreement between the network output and the truth map. From table 6.22 it is observed that the network maps 3038 pixels (approximately 4% of the area) as being saline as opposed to 3857 pixels (approximately 5% of the area) mapped by the technician.

Table 6.23 summarises the probability that given the network results, the technician mapped the area in the same way. For example, if the network mapped the area as $NotAffected$, then the probability that the technician mapped it the same way is 0.89. Similary, if the network mapped a pixel as $Saline$, then the technician was most likely to map it as $Saline$ (0.63) or $PotSaline$ (0.31) but not likely to map it as $NotAffected$ (0.06).

From figure 6.22 it is evident that each pixel label is chosen with a varying degree of certainty.

Table 6.24 summarises the conditional probability $P(Network|Technician)$. From the table, it is observed that if the technician mapped an area as $NoRisk$ then the network was almost certain to map it the same way. On the other hand, if the technician mapped an area as $SaltAffected$ then the network was almost equally likely to map it as $NotAffected$ or $Saline$. The reasons for this are the small size of the $Saline$ area (4% of the area) and the fact that the network slightly underestimates the extent of salinity.

The areas identified as being saline in figure 6.18, but not in figure 6.21, were not part of the broad valley systems and supported groundcover in one or more of the three years considered. Given the error rates associated with the use of Landsat TM for spectral landcover interpretation, the system interpreted these areas as $Normal$ with a proportion of $PoorProductivity$.

Another aspect of combining data using Bayesian networks is that they provide the basis for probabilistic expert systems. A user of such a system can work interactively with the network to test hypotheses and examine what was learned by the system. For example, the prior probability of salinity is $P(Salinity) = 0.06$, which indicates that before any other evidence is added it would be assumed that

6% of the area is saline. If the user enters the evidence $LF = BroadValley$, then the revised figure is $P(Salinity) = 0.33$, indicating a five-fold increase in our belief for salinity. The label *Normal* is still more probable at this point. Then if the user enters an observation of $LC1 = Salt$, the updated figure becomes $P(Salinity) = 0.52$, and the label *Salinity* is chosen, but with low certainty as $P(Normal) = 0.48$. If $LC2 = Salt$ and $LC3 = Salt$, then the label *Salinity* is chosen with more confidence, with $P(Salinity) = 0.85$.

Other relationships were also learned, for example, if the user enters evidence $S = DeepSand$, then $P(LCS = PoorProductivity) = 0.12$, whereas if the user entered $S = Other$, then $P(LCS = PoorProductivity) = 0.04$, indicating that *PoorProductivity* is three times more likely on *DeepSand* than on *Other*. Further, *PoorProductivity* is 1.5 times more likely for *DeepSand* on *Hilltops* than for *DeepSand* alone.

### 6.5.8 Conclusions

Bayesian networks provide a suitable modelling environment for constructing a knowledge–based system for the application considered here. Such a system can combine multiple sources of data to produce better land condition classifications and can meaningfully indicate to the user the knowledge from which the results were obtained.

When much data are available, extracting exact specifications of rule weights from experts may not be required, as techniques such as the EM algorithm can be used in some instances to estimate the rule weights from the data. In this way, experts can impart their knowledge to the modeller in a more conceptual manner and be saved the task of trying to rigorously specify their knowledge in terms of system parameters.

Expressing uncertainty in terms of probabilities is not considered a disadvantage in this application, since interpreting probabilities as relative frequencies of counts provides the user with a tangible interpretation of the numbers the system

provides.

One of the deficiencies of the model is that the system only embodies a subset of the relationships cited by the experts. This was due to the necessary data not being available for this region.

## 6.6 Summary

When adequate classification results cannot be obtained using a single source of data, then results can generally be improved by combining a number of different sources of data. This chapter considered a number of experiments for combining multiple sources of data for the purpose of monitoring land condition/cover in the West Australian wheatbelt.

Previous research by Wallace and Wheaton (1990); Wheaton et al. (1992) demonstrated that single-date remotely sensed Landsat TM data could be used to map salt-affected land in the West Australian wheatbelt, although some errors in the classifications were reported. The research was conducted in and around the town of Bindi-Bindi. MLC was used to perform the classification.

Building on the research of Wallace and Wheaton (1990); Wheaton et al. (1992), this chapter described a series of experiments in combining data using Bayesian networks, ranging from networks based upon relatively simple independence assumptions to more complicated structures.

Firstly, a series of simple models for combining ground data with a single source of remotely sensed data were tested and compared with MLC. In the tests, soil data were used as ground data. The models were applied to data from four consecutive years in order to see if improvements could be consistently obtained. Each model was compared for its accuracy in salinity mapping and paddock-use mapping. The spatial consistency of the resulting interpretations was also considered.

The first model, called MODEL1, was originally proposed by Strahler (1980) and uses the ground data to influence the prior probability of observing a particular

landcover. The second model, called MODEL2, is similar to MODEL1 except that the classification error rates obtained from the MLC are incorporated. Both models could be considered simple extensions of the MLC.

An EM algorithm was used to estimate the model parameters. The EM algorithm was appropriate, as limited ground truth data were available. In these examples, spectral signatures had been previously estimated from a series of sample sites sparsely located over the MKG study area. The EM algorithm was used to estimate the missing model parameters based on what could be inferred from the spectral component of the model.

Both MODEL1 and MODEL2 significantly improved the accuracy in mapping salinity and also the spatial uniformity of the resulting interpretation. On average, MODEL1 improved the accuracy by approximately 4% and MODEL2 by 8%. Further, it was demonstrated that estimating the model parameters using the EM algorithm improved results by some 5% as compared with a more obvious estimation procedure. MODEL1 essentially performed the same as the MLC for paddock-use mapping; however, MODEL2 adversely affected the accuracy of mapping paddock-use.

The reason for the improved salinity mapping accuracies was that salinity is associated with particular soil types and not associated with others. That is, there is a useful relationship between salinity and soil type. The opposite was generally the case for paddock-use.

The reason MODEL2 had the most extreme preformances, performing the best for salinity mapping and the worst for paddock-use mapping, was that incorporating the classification error rates reduced the importance of the spectral component of the model in relation to the non-spectral component, allowing the soil data to have greater influence on the interpretation. The use of the EM algorithm further enhanced this. The lesson to be learnt was that if the ancillary data do not contain information relevant to the class of interest, then results may be adversly affected, as the model attempts to generalise what landcovers are associated with particular

ground classes, which in the case of paddock-use and soil data was sporadic.

Ideally, one would hope that salinity mapping and paddock-use mapping could both be improved by the same model. It was noted in the use of MLC, MODEL1 and MODEL2 that errors associated with salinity were located in different positions for the sequence of four years considered. It is known that salinity should be temporally quite stable, given the relatively short time period considered: temporal rules promised one means for reducing the errors. Further, cropping rotations would tend to make some paddock-use types more probable than others in any particular year. For example, wheat is typically planted in the season following a crop of lupins. In this way temporal rules also promised to improve paddock-use mapping.

In section 6.4, the effect of temporal rules was explored. Ground data such as soil types were ignored so that the effects of the temporal rules could be observed. Two models, called MODEL3 and MODEL4, were conceived. MODEL3 and MODEL4 assumed that the landcover observed at year $i$ is conditionally independent of all other years given hard evidence of the land cover for the years $i-1$ and $i+1$. MODEL3 and MODEL4 temporally link the interpretations obtained for the years 1989, 1990, 1991 and 1992.

With these slightly more complicated models, a point to note is that with the probability propagation scheme used, the resulting interpretations are conceived jointly, that is, what is observed for any one year can affect the interpretation of all the other years. This approach should not be confused with older applications of Bayesian inference, such as used by Aspinall (1992); Bonham-Carter et al. (1988), which performed calculations in the direction of the network arrows. Further, the older schemes did (could) not accommodate estimation schemes such as the EM algorithm, which requires joint marginal probabilities to be evaluated.

MODEL3 and MODEL4 are basically the same, except that like MODEL2, MODEL4 incorporates the classification error estimates derived for the spectral component of the model. The models were tested against the same criteria as

MODEL1 and MODEL2. Again, because of limited ground information, the EM algorithm was used to estimate the model parameters.

On average, MODEL3 improved the salinity mapping by 3% and paddock-use mapping by 2% as compared with the MLC. MODEL4 performed better, improving salinity mapping by 10% and paddock-use mapping by 3% as compared with the MLC. Again, while the gains were rather modest, improvements were consistently obtained for each of the years. Further, little effort was expended in obtaining the improvements.

Improvements in the spatial uniformity of the resulting interpretations accompanied the improved accuracies. These improvements were partially the result of the temporal rules correctly labelling isolated pixels which were misclassified as salt-affected by the MLC. MODEL3 improved the spatial uniformity by roughly 10% as compared with the MLC and MODEL4 by roughly 13% as compared with the MLC.

Also considered in section 6.4 was the incorporation of neighbourhood operations into the models. The reason for this was that while the temporal models improved the spatial uniformity of the resulting interpretations, isolated misclassified pixels were still produced. Explicitly including the spatial relationships of a pixel label with that of its neighbours appeared advantageous.

The approach devised extends the ideas of Besag (1986), who considered neighbourhood operations in the classification of a single image source. Noting that models such as MODEL3 and MODEL4 simultaneously produce landcover interpretations for multiple years, simultaneously incorporating neighbourhood operations for each of the years appeared advantageous. A new scheme that may be applied to general Bayesian networks is proposed. To demonstrate the approach, neighbourhood operations were added to MODEL3 and MODEL4 to produce two new models, called MODEL5 and MODEL6 respectively.

MODEL5 and MODEL6 were tested against the same criteria as all previously considered models. Also, for comparison purposes, the neighbourhood model pro-

posed by Besag (1986), which is the MLC with neighbourhood operations included and called MLC-NBR here, was applied to the same data.

The MLC-NBR consistantly outperformed the MLC, reducing the salinity mapping errors an average of 3% and improving the paddock-use mapping by an average of 4%. This evidence suggested that neighbourhood operations are useful at an operational level.

On average, MODEL5 outperformed the MLC, the MLC-NBR, and its non-neighbourhood couterpart MODEL3. On average MODEL6, outperformed all other models. On average, MODEL5 reduced the salinity mapping errors by 5% and MODEL6 by 15% as compared with the MLC, and 2% and 12% as compared with MLC-NBR. Comparing MODEL3 with MODEL5, and MODEL4 with MODEL6, it was observed that incorporating neighbourhood operations reduced the salinity mapping errors by 2% and 5% respectively, and increased paddock-use mapping accuracy by 4% and 3% respectively.

In examining the results of MODEL6, it was noted that the mapping of severely salt-affected land was not as consistent (temporally) as expected. Examination of those temporal parameters of MODEL6 which were estimated by the EM algorithm revealed that poor (salinity) spectral information for 1990 resulted in a poor estimate for the probability of salinity in 1991 given that observed in 1990, reducing the temporal consistency of the model for this outcome. To remedy this poor estimate, the parameter $p(SaltSure_i|SaltSure_{i-1})$, $i = 1990, 1991, 1992$, were subjectively initialised to be close to unity, and then the EM algorithm applied to re-estimate the parameters for use with MODEL6. MODEL6 with the new parameters, called MODEL6*, was then used to interpret the data. MODEL6* produced an average of 3% less errors than MODEL6 for mapping salt-affected areas. Paddock-use mapping accuracy was largely unchanged.

Comparison of MODEL6 and MODEL6* demonstrated that prior knowledge of the likely values (weights) of the model parameters (rules) could be advantageous when applying an estimation technique such as the EM algorithm. These weights

could be derived from previous experience obtained from similar applications in other areas or provided by an expert.

In section 6.5, the specific problem of mapping salt-affected land was considered in greater detail, with the aim being to produce interpretations similar to what a human expert would provide. A knowledge-based approach was taken. Knowledge gained from consultation with experts was used to define the relationships that were likely to exist, that is the structure and prior rule weights of the Bayesian network, and what remote sensing and GIS data would be useful for mapping salinity.

As in the earlier models, the EM algorithm was used to estimate the rule weights, although more emphasis was placed on expert knowledge to define initial weights. This resulted in the salinity interpretation as derived from the network more closely resembling (compared with the models considered earlier) that provided by the expert.

Also demonstrated in section 6.5 was the basic belief revision aspects of Bayesian networks when used interactively to investigate *what-if* scenarios. This feature allows the network to be interrogated before application to a new area say, or mistakes produced in an application being better able to be readily identified and understood.

# Chapter 7

# Application — Predicting Salinity

## 7.1 Introduction

In much of the Western Australian agricultural region, the clearing of land has lead to rising saline groundwater, resulting in the loss of previously productive land to salinity. Based on farmer surveys conducted in 1979 and 1989, the Australian Bureau of Statistics reported that 443000ha (2.8%) of previously arable land was lost to salinity, at a rate of about 18000ha each year.

Aerial photo interpretation has been used to assess the extent of salinity in some areas (Nulsen, 1981), but this is time consuming and expensive. More recent efforts (Wallace and Wheaton, 1990; Wheaton et al., 1992, 1994) have used Landsat TM data to estimate the extent of salinity. Typically, statistical approaches such as maximum likelihood classification are used to classify the data into different landcover classes, which are then associated with the land being severely affected, slightly affected and not affected by salinity.

The above approaches provide a means of monitoring the extent of salinity, but provide no insights into the likely effects that remedial actions such as tree planting will have on the problem, or given the current land status, what areas are at risk of salinisation in the longer term. The latter is imperative for the formulation of remedial measures.

To explore ways of achieving this, this chapter considers the development of a Bayesian network which may be applied to mapping and predicting salinity. The area of focus is the Upper-Kent River Catchment introduced in section 3.3.

Of interest in this research is to examine to what degree Bayesian computer-based reasoning models can reproduce salinity mapping and predictions made by experts, with the former having the potential to provide cost effective and reproducable results.

The experts considered a period of ten years to be a suitable time frame for making predictions, and (available) data considered important for the task were collected for three time periods in the 1970s, 1980s and 1990s respectively. The data collected are discussed in section 3.3.1.

The methodology of the approach was:

1. Consult with experts to gain knowledge of the problem.

2. Based on the expert knowledge, obtain data (where possible) that are relevant, possibly with further processing, to applying the knowledge over the whole catchment.

3. Based on expert knowledge and available data sets, construct the knowledge-based system.

4. Use the resulting model to produce salinity risk maps for the region.

5. Validate the results by comparing the system output to independent validation data, and if acceptable, form a catchment salinity summary.

Preliminary knowledge of the process of predicting salinity resulted from a forum of experts held in 1994. The forum attempted to resolve conflicting views on the process of predicting salinity by a consensus approach.

Subsequent knowledge of the application domain was obtained from frequent consultation with experts who had detailed local knowledge of the catchment.

In section 7.4, the model devised from expert knowledge and available data is described. Based on the knowledge, variables were derived from the data described in section 2.5. Variables were derived from making use of the landform partitioning techniques developed in the MKG region and simple extensions of these ideas. Extraction of variables is described in section 7.3.

Application of the model to data is given in section 7.6, from which regional estimates of the historical spread of salinity and likely future extent of salinity are given. This work has subsequently been published by Caccetta et al. (1995b).

## 7.2 Knowledge Elicitation

Initial knowledge on the factors relevant for salinity prediction was obtained from a forum of experts. The scope of the forum was to identify factors that would be necessary for predicting salinity regionally, as well as factors that would need to be considered if the results were to apply nationally. The factors deemed important by the experts included:

- The geology of the region.
- The historical flushing of a region: on a simplistic level, the amount of salt stored in the sub-surface profile is reduced for a particular region if forces such as rainfall and ground water movement have dissolved and moved it to another region.
- The regional rainfall, which itself influences the amount of historical flushing. For example, regions with annual rainfall greater than 1100mm, or less than 200mm, were not considered to be at risk. The expected rate of salinisation for areas between these two extremes is dependent on the amount of rainfall, with salinity expressing itself in higher rainfall areas more quickly than in areas with lower rainfall.

- Points of convergent flow: the rate of groundwater rise is increased when two or more flow paths meet.
- Landform patterns and associated landform elements.
- Groundwater salinity.
- Land use: for example, unlike traditional grazing practices, tree farming can have the effect of drawing down the water table.
- Amount of clearing: areas that have been extensively cleared are at greater risk than those where little clearing has occurred.
- The subsurface geology: subsurface structures that inhibit subsurface water flow increase the risk of salinisation.
- Time since clearing: once the land has been cleared, water tables typically start to rise and at some period will rise close to the land surface.
- Rate of groundwater rise.
- Changes in hydraulic conductivity, including changes in soil types and placement of roads: barriers which force saline ground-water to the surface, from where it evaporates, will result in increased concentrations of salt.

Clearly, some of these factors are interrelated, and some data which are relevant for assessing the significance of some factors were not available.

The strategy adopted was, where possible, to derive variables that provide estimates for the factors in question. For example, classification of historical Landsat data provides information on land use and the amount of clearing. The experiments conducted in chapter 5 provided the basis for factors relating to surface morphology. The application of these ideas to the Kent catchment is now described.

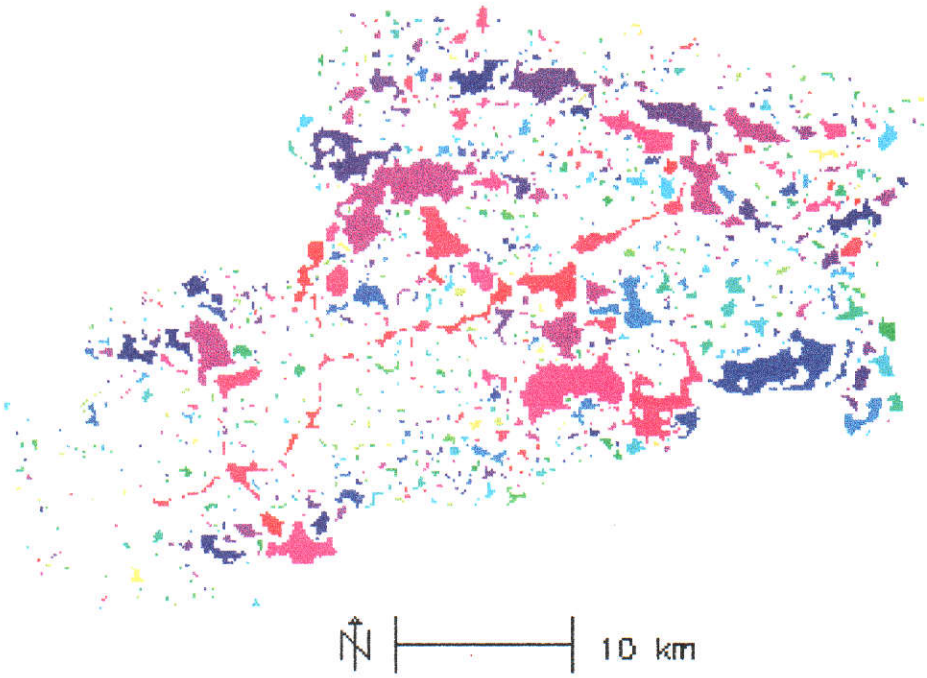



Figure 7.1: Result of INFLOWS algorithm. The figure is coloured red, which indicates a high INFLOWS value, through pink, purple, blue, cyan, green and then yellow, which indicates a low INFLOWS value. Pixels that are white are not flat.

## 7.3 Extracting Landform Features and Related Variables

The algoithms INFLOWS, ALLFLOW3 and FLOWSLOPE were applied to the DEM data to form a flat-area stratification, an *upslope area* map and a *flow-slope* map respectively.

The upslope area map was stratified into five classes which become surrogates for landform types, defined as *HillTops*, *Ridges* and *UpperSlopes*, *UpperValleys*, *LowerValleys* and *BroadValleys*.

The results of the INFLOWS and FLOWSLOPE operations are shown in figures 7.1 and 7.2 respectively. The stratified ALLFLOW3 map is shown in figure 7.3.

Convergent flow points were extracted from the valley system classes of the stratified ALLFLOW3 image. For display purposes, a buffer zone around these points

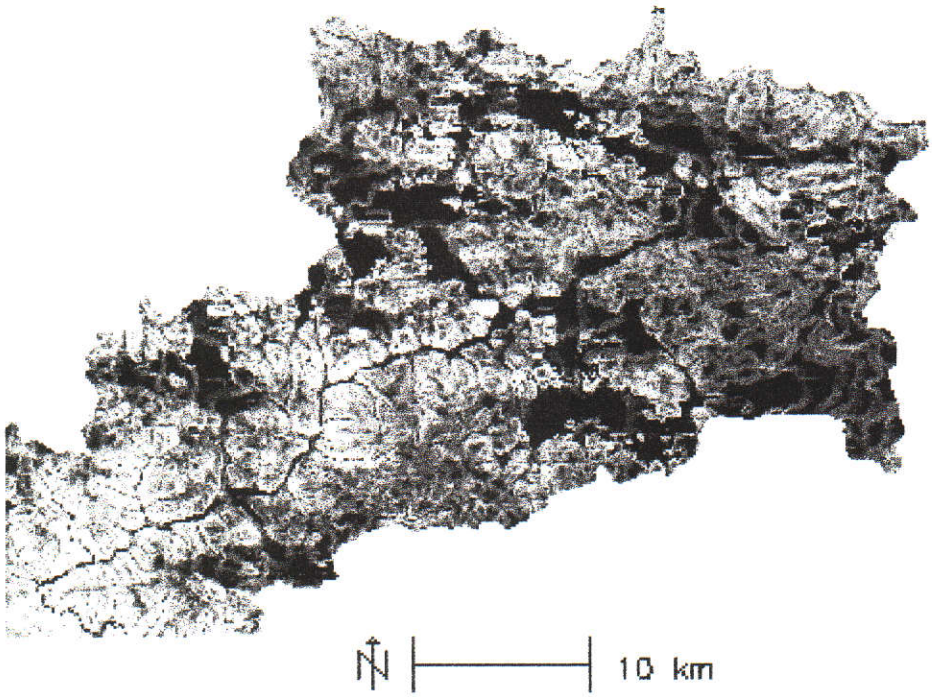



Figure 7.2: Result of the **FLOWSLOPE** algorithm in greyscale. White and black indicate high and low slope respectively.

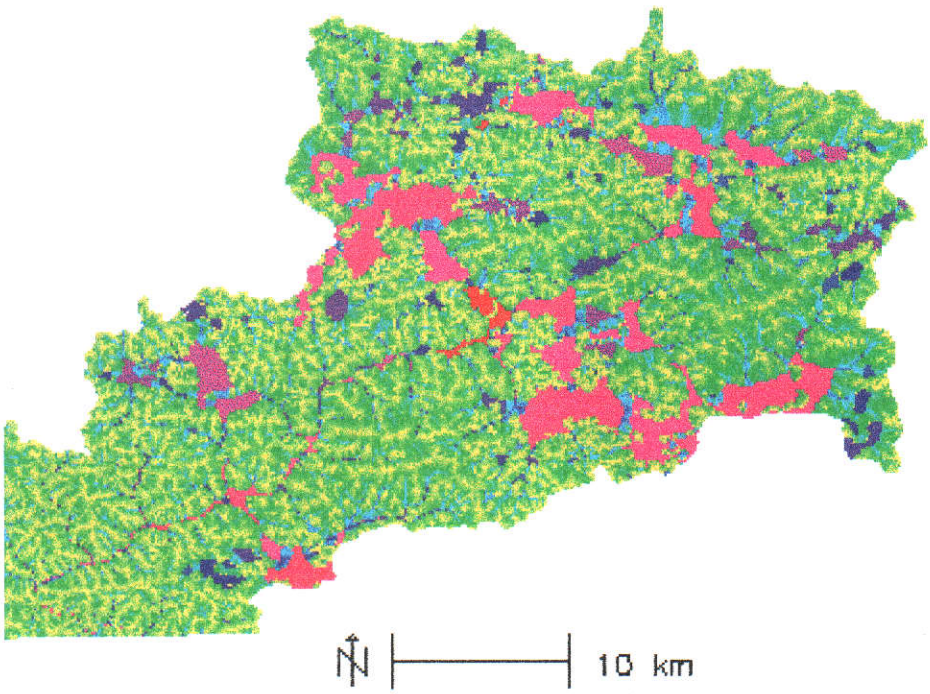



Figure 7.3: Result of the ALLFLOW3 algorithm. The figure is coloured red, which indicates a high accumulation value, through pink, purple, blue, cyan, green and then yellow, which indicates a low accumulation value.

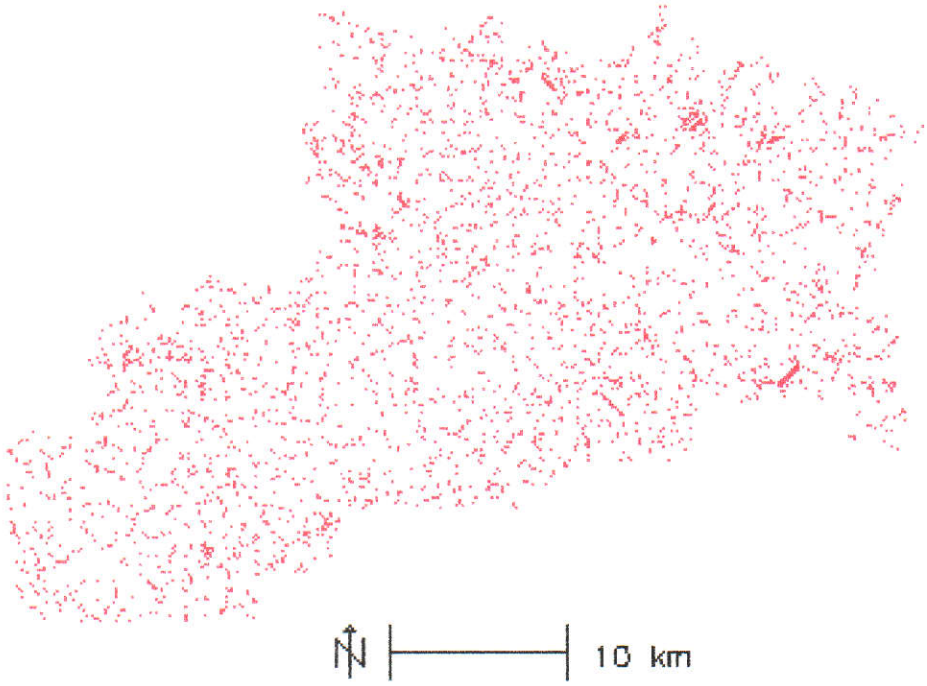



Figure 7.4: Buffered Convergent Flow Points.

was defined and the results shown in figure 7.4.

Visual inspection of the points against the training data indicated that these points would be of limited value for predictions in this region, as valley systems, as opposed to small/isolated sections of valley systems, were saline. That is, although it may be possible that the points of convergent flow are more at risk of salinity and hence go saline first, at the time the observations of salinity were made, the rest of the valley system was also salt-affected. This observation appears to be consistent with the expert knowledge that the rate of salinisation is greater in higher rainfall zones; as in the lower rainfall area of Xantippe, mainly points of convergent flow were salt-affected. For this reason, points of convergent flow were not considered further as their effect could not be quantified.

The classification of Landsat data provides information on land that has been cleared. This information may be combined with the `ALLFLOW3` algorithm to provide estimates of the amount of land above any location that has been cleared.

The estimate, called *upslope cleared area (UpSlpClr)* in the following, was formed by the following method:

- An image is formed where each pixel identified as being cleared is given a value of one, all regions with remnant vegetation a value of zero.
- The `ALLFLOW3` algorithm is applied using the DEM and the above image as the rainfall event, resulting in an image representing *upslope cleared area.*

A rule-of-thumb commonly quoted by experts is that if less than 70% of a (sub)catchment has been cleared, then the catchment is not at risk of salinisation. A combination of the upslope area and upslope cleared area yields an image of *percentage upslope cleared area* which may be useful in quantifying this belief.

The images, *upslope cleared* and *percentage upslope cleared*, are shown in figures 7.5 and 7.6 respectively.

Another factor cited as important by the experts is the salinity of the local ground water. Detailed data describing the spatial variation of this variable was not available. Based on work in the Denmark and Kent catchments, the near surface ground water salinity is related to the landform pattern of the region (Ferdowsian and Greenham, 1992; Ferdowsian, 1993, 1994). Based on this work, a surrogate for ground water salinity was generated from the landform pattern map, and is depicted in figure 7.7.

## 7.4 Proposed Model

Based on the knowledge gained from experts, available data and visual examination of data, a Bayesian network model for predicting salinity was conceived; its structure is depicted in figure 7.9.

The network uses estimates of *upslope cleared area*, *percentage upslope cleared area*, *flowslope*, *land cover*, *ground water salinity* and historical salinity risk to map and predict salinity. The variables *flowslope* and *ground water salinity* are treated as being stationary and the other variables as dynamic.

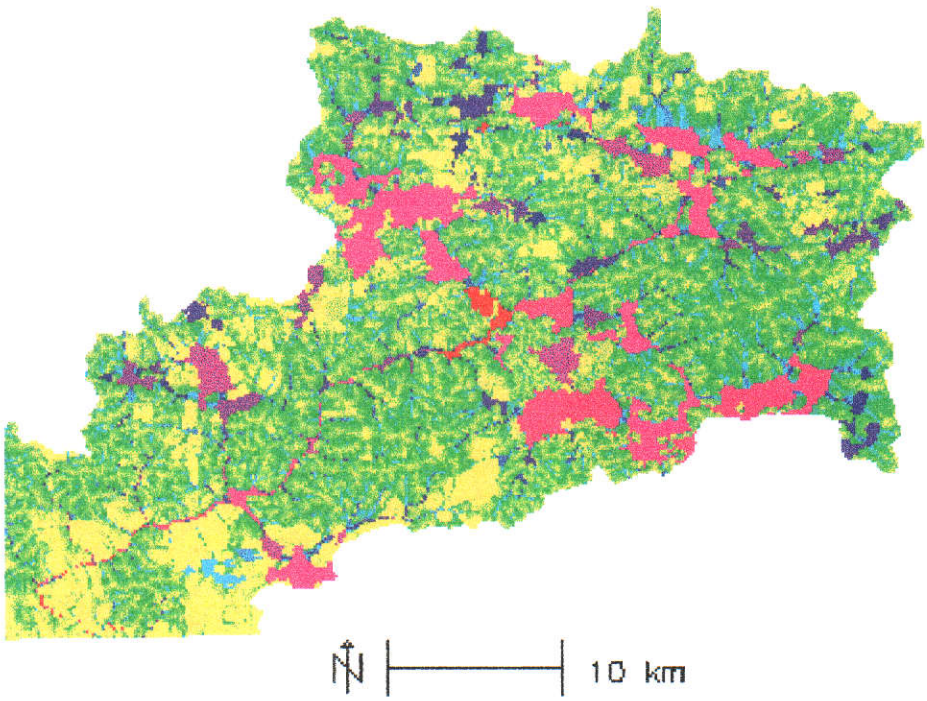



Figure 7.5: Upslope Cleared area. The figure is coloured red, which indicates a relatively large upslope cleared value, through pink, purple, blue, cyan, green and yellow, which indicates a relatively low upslope cleared value.

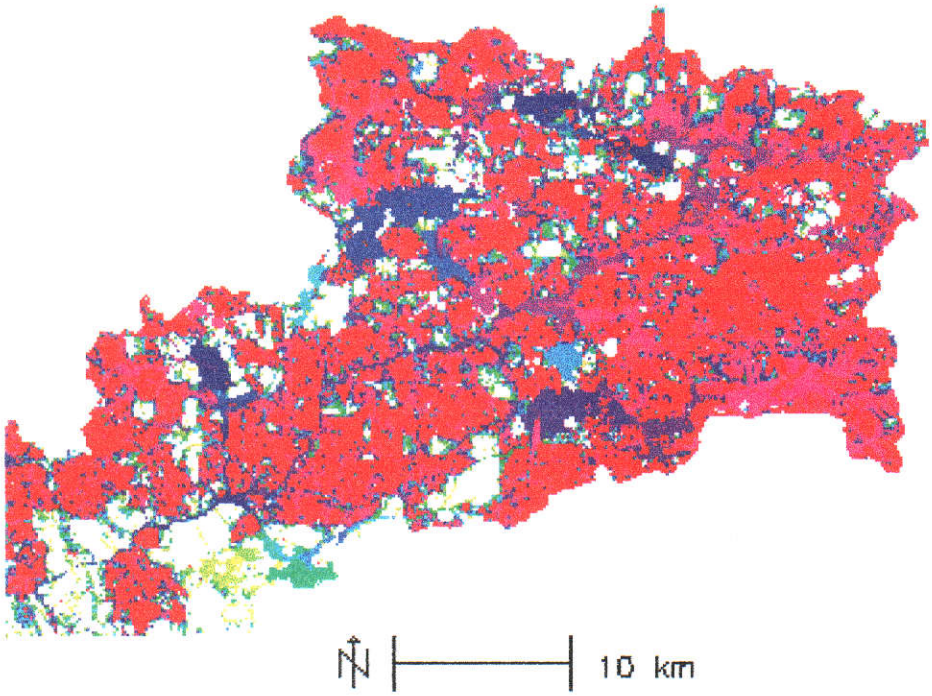



Figure 7.6: Percentage upslope cleared. The figure is coloured red, which indicates 100% upslope cleared, through pink, purple, blue, cyan, green, yellow and white, which indicates a minimal percentage of upslope clearing.

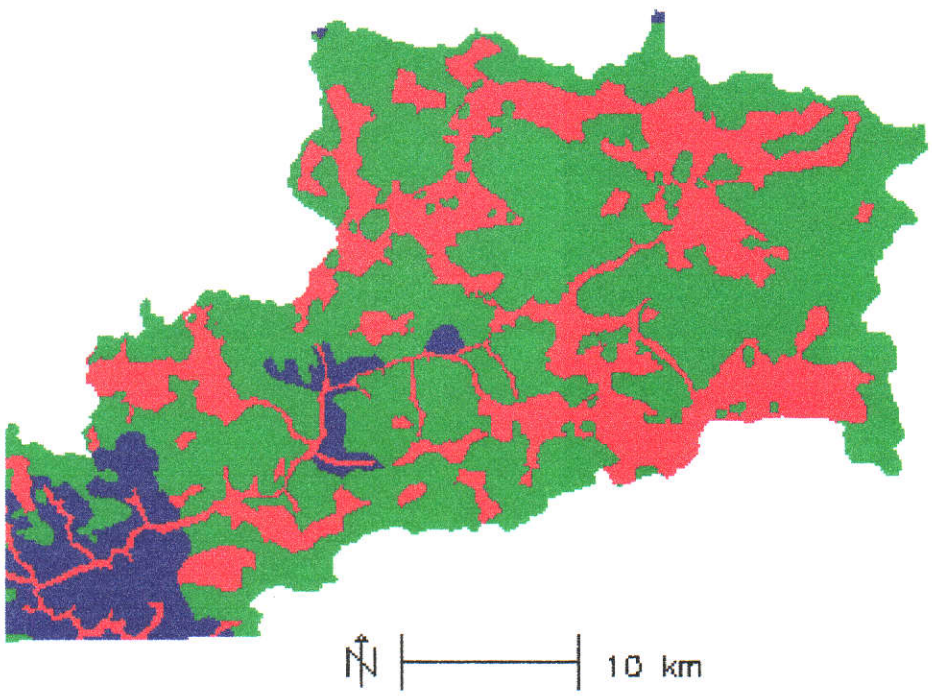



Figure 7.7: Expected Ground–Water Salinity. Red, green and blue regions have a high, medium and low expected groundwater salinity respectively.

In figure 7.9, the vertex labels represent the above variables as follows:

- UpSlpClrXX — upslope area cleared for year XX.
- FlowSlope — flowslope, slope in the direction of flow.
- PerClrXX — percentage upslope cleared for year XX.
- GndWatSal — (expected) ground water salinity, as derived from landform patterns.

The variables *DischXX* and *LndCndXX* represent the latent variables *potential for discharge in year XX* and *land condition status for year XX* respectively.

The variables have the following states:

- $LndCndXX = \{NoRisk,\ PotSaline,\ Saline\}$
- $DischXX = \{Yes,\ No\}$

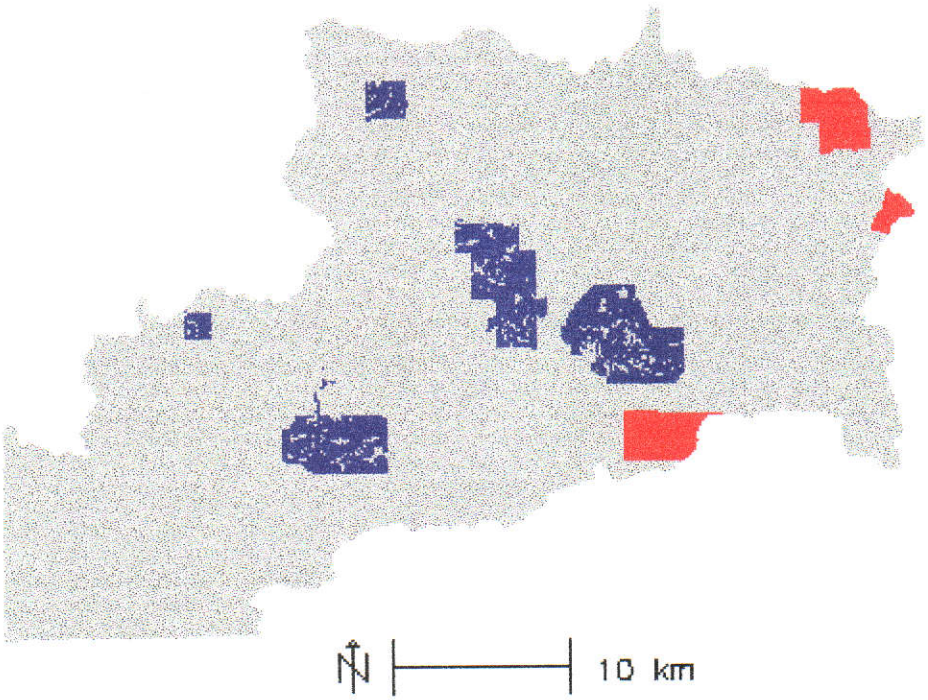



Figure 7.8: Location of training and validation data, coloured blue and red respectively

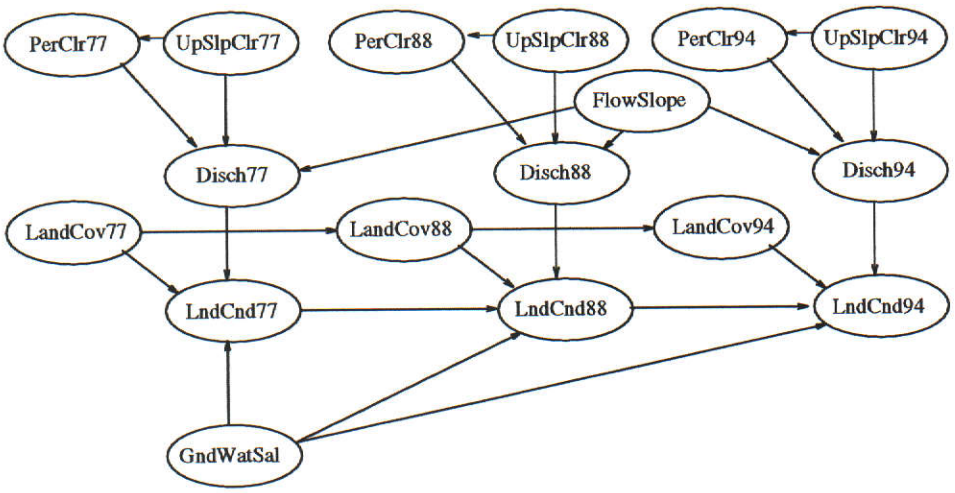



Figure 7.9: Proposed model.

- *FlowSlope* = {(0,2), (2,4), (4,6), (6,8), (8,10), (10,12), (12,14), (>14)}
- *GndWatSal* = {*Low, Medium, High*}
- *LndCovXX* = {*Water, RemVeg, BareSoil, BareSalt, WaterSideRemVeg, AgrLand*}

The model has the following interpretation:

> *Based on the total area cleared, the percentage of the area cleared and the flowslope of the land, an area has a certain risk of discharging ground water, which may or may not be saline. Given a general risk of an area developing a discharge, landcovers may influence the risk of salinity or determine that an area is already affected by salinity. This risk is further influenced by the historical salinity risk of the area and the ground water salinity.*

Unfortunately, because of lack of information on historical landuse, error estimates (such as those considered in section 6) for the classifications derived from the Landsat data were not available for this region and hence were not included in the model.

## 7.5 Estimating the Network Parameters

As error estimates for variables were lacking, there appeared little advantage in using the EM algorithm to estimate the model parameters, and hence the OBVIOUS technique given in section 6 was used.

The training data were used in the following way: for the given time period, areas mapped by the expert as *not affected by salinity* in the next time period were considered not presently at risk; areas mapped by the expert as *salt-affected* in the present time period were considered salt-affected; those areas that were not mapped as *salt-affected* in the present time period, but subsequently mapped as

salt-affected in the next time period were considered as areas that were potentially saline in the current time period. In this way, the class *potentially saline* represents those areas that will go saline in the next time period.

To estimate the parameters, the data were mapped to the network variables as follows:

- The training data provided by the expert were mapped to the variables DischXX and LndCndXX. For DischXX, the experts' *PotSaline* and *Saline* classes were grouped to form the DischXX = *yes* class. The experts' *NoRisk* class formed the DischXX = *no* class.
- The classifications derived from MLC of the Landsat TM data were mapped to their respective LandCovXX variables.
- The data representing the expected ground water salinity were mapped to the variable GNDWatSal.
- The FlowSlope data were mapped to the variable FlowSlope.
- The PerClrXX data were mapped to their respective PerClrXX variable.
- The UpSlpClrXX data were mapped to their corresponding UpSlpClrXX variable.

The *obvious* approach, as described in section 4.5, was then used to estimate the parameters.

## 7.6 Results

### 7.6.1 Computational Aspects

The data were processed using the computational scheme which maintains the results of previously processed data combinations (see section 4.4). The processing scheme reduced computational time to about a third of the time required by the

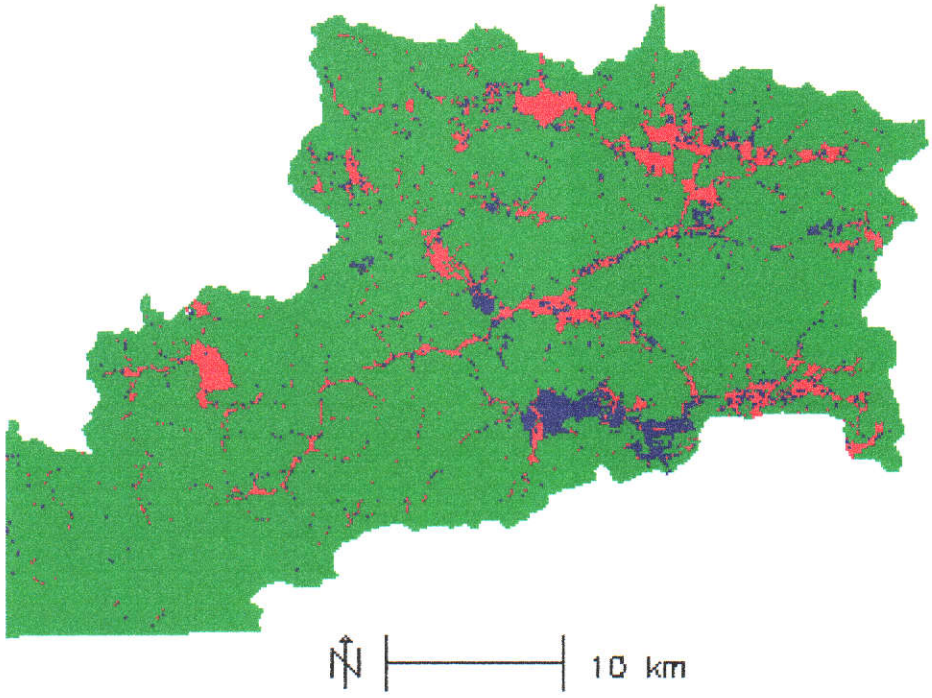



Figure 7.10: Network output: most likely class label, 1988. *Saline*, *PotSaline* and *NoRisk* areas are coloured red, blue and green respectively.

method which does not use the binary search tree. The area considered had dimension 1934 by 1666 pixels and took approximately 56 hours on a SUN SPARC 10.

Probabilistic output maps for the nodes $LndCnd77$, $LndCnd88$ and $LndCnd94$ were produced, along with labels of the most likely outcome for each variable. An example of the system output is given in figure 7.10, which depicts the most likely class label for the year 1988.

### 7.6.2 Comparison with Training Data

The results were firstly compared with the original training data; the mapping accuracies for each of the years are summarised in table 7.1. In table 7.1, the integer numbers represent pixel counts and the number in brackets represent the probabilities $P(Expert|Network)$ and $P(Network|Expert)$ depending upon whether the

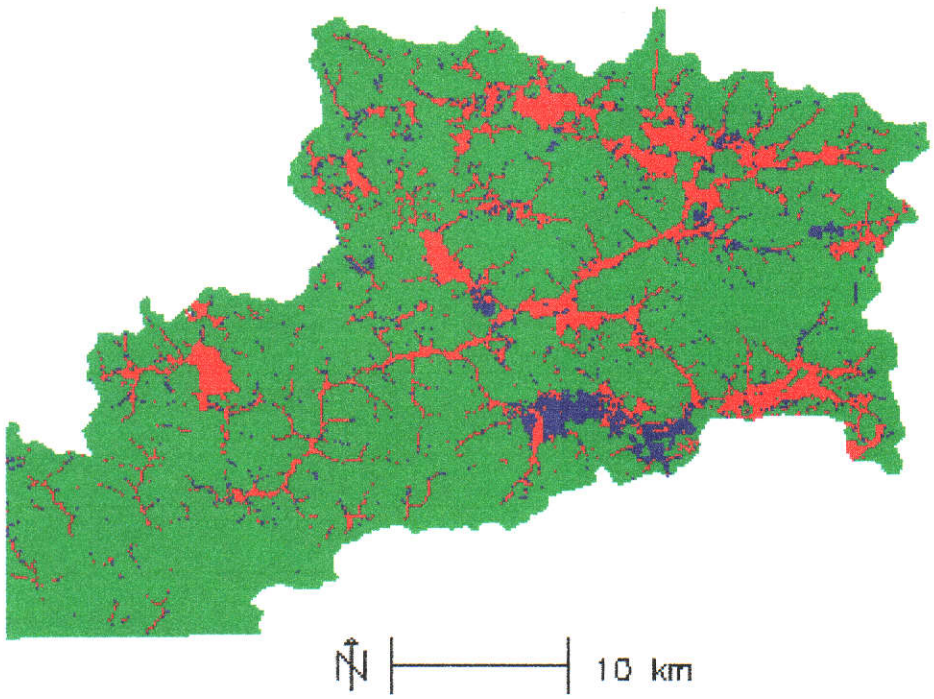



Figure 7.11: Network output: neighbour-modified results 1988. *Saline*, *PotSaline* and *NoRisk* areas are coloured red, blue and green respectively.

| Expert | Network | | | |
|---|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* | Total |
| | Year - 1977 | | | |
| *NoRisk* | 71372 (**0.92**) | 741 (**0.40**) | 797 (**0.18**) | 72910 |
| *PotSaline* | 2648 (**0.03**) | 359 (**0.19**) | 934 (**0.21**) | 3941 |
| *Saline* | 3662 (**0.05**) | 765 (**0.41**) | 2644 (**0.61**) | 7071 |
| Total | 77682 | 1865 | 4375 | 83922 |
| | Year - 1988 | | | |
| *NoRisk* | 63770 (**0.88**) | 673 (**0.24**) | 1600 (**0.20**) | 66043 |
| *PotSaline* | 4858 (**0.06**) | 738 (**0.26**) | 1271 (**0.15**) | 6867 |
| *Saline* | 4222 (**0.06**) | 1413 (**0.50**) | 5377 (**0.65**) | 11012 |
| Total | 72850 | 2824 | 8248 | 83922 |
| | Year - 1994 | | | |
| *NoRisk* | 51053 (**0.79**) | 1808 (**0.32**) | 2659 (**0.19**) | 55520 |
| *PotSaline* | 7829 (**0.12**) | 1150 (**0.20**) | 1544 (**0.11**) | 10523 |
| *Saline* | 5661 (**0.09**) | 2701 (**0.48**) | 9517 (**0.70**) | 17879 |
| Total | 64543 | 5659 | 13720 | 83922 |

Table 7.1: Mapping accuracy over training data.

numbers are shown in bold or normal text respectively. From table 7.1, we observe that for 1977, the network overestimated the amount of land that was not at risk while underestimating land that was potentially at risk or already salt-affected. This also occurred in the other two time periods.

Although the network appears to have underestimated the extent of salinity, from table 7.1 we observe that for 1977, 92% of the pixels mapped as *NoRisk* by the network were not at risk and that 61% mapped as *Saline* were salt-affected, indicating that although the classes were underestimated, the general locality of the classes was correct. The results for the potentially saline class, however, were not good. In 1977, what was mapped as potentially saline by the network was land that was predominately already saline or not at risk for that time period. For the other periods, what was mapped as potentially saline was typically already considered salt-affected.

The confusion between the classes *PotSaline* and *Saline* can be explained by

the following points:

- Analysis of the Landsat data (late winter/early spring) (Evans, 1995a) showed that no spectral discrimination existed between salt-affected pastures, non-affected pastures and pastures in areas at risk. This was also the case for areas mapped by the expert as salt-affected and non-salt affected remnant vegetation. The only salt class that was spectrally separable was severely (salt)scalded land. For this reason, in the salt-affected regions supporting pastures and remnant vegetation, the network essentially must guess whether the area is already affected or yet to be affected.

- The historical spread of salinity was mapped by the expert from historical aerial photographs. Given the definition of *Saline*, we would expect an element of error in the boundaries of what was mapped as SaltAffected and NotAffected from this process and hence errors in the derived *PotSaline* class used in training the network.

Consultation with the experts indicated that Landsat TM data from the dates October, November or December may provide information about the state of pastures. The reason for this is that as the pastures die off toward summer, pastures in salt-affected and potentially affected areas will take a little longer to die, as they have access to the relatively high water table associated with these areas. Unfortunately, no Landsat data were available within these months for several years due to cloud cover, and this information could not be incorporated.

The reason for the network underestimating the extent of salinity is largely due to the following reasons[1]:

- Much of the salinity missed by the network was located at the foothills of land adjacent to *Saline* valley systems. No variable identifying a foothill exists in the model. Given the results from chapter 5, extracting foothills

[1]These shortcomings in the data would pose a problem for any form of classifier.

with DEM data derived from contours would be unreliable, especially in flatter areas of terrain.

- The above problem is compounded by the lack of discrimination from Landsat TM in salt-affected pastures.

As what the network maps as *Saline* is generally a subset of the area considered by the expert as *Saline*, neighbourhood approaches that have the effect of expanding the mapped areas appear useful. However the neighbourhood technique introduced in section 6.4.2 is inappropriate for this task as it produces a *clustering* effect and not a means of expanding the region.

Therefore to gain a feel for what effect expanding the region may have, the following *ad hoc* operations were performed:

1. For each time period, those areas mapped by the network as *Saline* or *Pot-Saline* were extracted.

2. Making use of the DEM, a `HeightAbove` operation was performed using the regions extracted in the previous step as the features of interest. This operation produced (a) an image indicating the height that a pixel is above the closest feature pixel. and (b) the distance each pixel is from the nearest feature pixel

Making use of these images, the class labels of training data for 1994 within $X$m. and $H$m. (in elevation) of the features were examined and are summarised in table 7.2. In table 7.2, the numbers in each cell (from left to right) represent the counts of *NoRisk*, *PotSaline* and *Saline* pixels respectively as mapped by the expert which are adjacent to *Saline* pixels as mapped by the network. From the table, it is observed that the majority of pixels adjacent to those areas mapped as *Saline* by the network are also saline. That is, including these neighbours as $(Pot)Saline$ in the final classification would give more right answers than wrong answers. Using this rationale, all *NoRisk* pixels adjacent to, and less than 3m. above, $(Pot)Saline$ areas mapped by the network were reclassed as $(Pot)Saline$.

| HeightAbove | DistanceFrom | | | |
|---|---|---|---|---|
| | [0,30m) | [30,60m) | [60,90m) | [90,120m) |
| [0,1m) | 5308,2900,12425 | 2279,811,1086 | 1579,371,451 | 4840,888,1186 |
| [1,2m) | 127,191,257 | 442,326,250 | 511,205,134 | 553,146,113 |
| [2,3m) | 20,37,24 | 238,212,156 | 310,120,44 | 353,120,44 |
| [3,4m) | 403,4,2 | 150,136,79 | 207,89,25 | 272,66,17 |

Table 7.2: 1994 *Saline* training sites within $X$m and $H$m height of network prediction.

The accuracy of the new classification map was then compared with the training data, and the results are summarised, in the same manner as for table 7.1, in table 7.3.

From tables 7.1 and 7.3, we observe that the total number of pixels mapped as belonging to each class more closely resembles the number identified by the expert than the output without the neighbourhood modification. For example, in 1977, estimates obtained from the training data indicate that 72910, 3941 and 7071 pixels had label *NoRisk*, *PotSaline* and *Saline* respectively, as compared with the network estimates of 72443, 3729 and 7750 for the same classes. For the years 1988 and 1994, the results were similarly well matched.

### 7.6.3 Comparison with Validation Data

Next, the accuracy of the network was compared against independent validation data. The data were mapped in the same way (by the same person) for three different areas in the catchment. The extent of salinity was mapped for the years 1973, 1985 and 1994, from which those areas mapped as *Saline* in 1985 and not *Saline* in 1973 were used as the *PotSaline* class for 1973. Similarly, a *PotSaline* class was constructed for 1985.

Classification error rates for the unmodified system output were generated, and are summarised for each of the years in table 7.4. From the table, it is observed that the extent and predicted extent of salinity was severley underestimated in 1977. The network predicted very little land lost to salinity in this period. The

| Expert | Network | | | |
|---|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* | Total |
| | Year - 1977 | | | |
| *NoRisk* | 68338 (**0.94**) | 2030 (**0.54**) | 2542 (**0.33**) | 72910 |
| *PotSaline* | 1807 (**0.02**) | 505 (**0.14**) | 1629 (**0.21**) | 3941 |
| *Saline* | 2298 (**0.04**) | 1194 (**0.32**) | 3579 (**0.46**) | 7071 |
| Total | 72443 | 3729 | 7750 | 83922 |
| | Year - 1988 | | | |
| *NoRisk* | 60435 (**0.91**) | 1213 (**0.45**) | 4395 (**0.30**) | 66043 |
| *PotSaline* | 3420 (**0.05**) | 765 (**0.28**) | 2682 (**0.18**) | 6867 |
| *Saline* | 2523 (**0.04**) | 715 (**0.27**) | 7774 (**0.52**) | 11012 |
| Total | 66378 | 2693 | 14851 | 83922 |
| | Year - 1994 | | | |
| *NoRisk* | 46716 (**0.85**) | 3349 (**0.43**) | 5455 (**0.26**) | 55520 |
| *PotSaline* | 5467 (**0.10**) | 1928 (**0.25**) | 3128 (**0.15**) | 10523 |
| *Saline* | 2731 (**0.05**) | 2442 (**0.32**) | 12706 (**0.60**) | 17879 |
| Total | 54914 | 7719 | 21289 | 83922 |

Table 7.3: Neighbourhood modified results.

network predicted most salinity to occur within the 1988 to 1994 period. Within this period, the network mapped the *NoRisk* areas with an accuracy of approximately 85%. However, what the network mapped as *PotSaline* was already *Saline* according to the expert (46%), correctly identified by the network (33%) or at *NoRisk* (21%). The confusion between these different classes is consistent with the results obtained for the training data.

The most accurate mapping results were obtained for the year 1994, where 90% (59%) of what was mapped as *NoRisk* (*Saline*) by the network was also mapped as *NoRisk* (*Saline*) by the expert. The general estimate of the spatial extent of salinity was also in agreement, with the network estimating that 9305 pixels (24%) of the area is affected by salinity as compared with the expert mapping 9311 pixels (24%) as being affected by salinity.

Next, error rates for the neighbour-modified output were calculated and are summarised for each year in table 7.5. As compared with the unmodified results

(table 7.4), the network estimates of the general extent of salinity improved for the years 1977 and 1988 while being overestimated for 1994. That is, the network estimated that salinity affected 7%, 14% and 36% of the validation region for the years 1977, 1988 and 1994 respectively as compared with the corresponding estimates by the expert of 7%, 14% and 24%.

However, like the unmodified results, much error in the labels remained. For instance, the best results were obtained for 1994 where the network identified 14066 pixels as being salt-affected of which only approximately 50% actually were. On the other hand, of the 9311 pixels identified by the expert as being *Saline*, the network correctly identified 7040 (76%) of them. The accuracies of identifying the location of *Saline* or *PotSaline* land is worse for the other two time periods, with the network more likely to identify affected land as being *NoRisk*.

The errors which occur are not surprising given the limited information provided by the Landsat data and the possibility that landuse may have significantly changed in any given period.

Given the above comments, the network maps are beneficial for providing historical estimates of the extent of salinity and the potential location of salinity; however, the exact time period in which the salinisation of any specific site will eventually occur is in doubt. The next section provides historical summaries of the extent and predicted extent of salinity in the Kent River Catchment.

### 7.6.4 Catchment Summary

The results were used to form historical estimates of the extent of salinity in the Upper Kent River Catchment. A map of the historical/future spread of salinity is depicted in figure 7.12, where green, cyan and purple represent areas which were affected by salinity in 1977, 1988 and 1994 respectively. The red areas are expected to go saline by 2004 and the grey areas are not expected to be affected by salinity in the next decade.

The historical extent of salinity is summarised in tables 7.6 and 7.7. The

| Expert | Network | | | |
|---|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* | Total |
| | 1977 | | | |
| *NoRisk* | 31925 (**0.84**) | 295 (**0.73**) | 882 (**0.75**) | 33102 |
| *PotSaline* | 3471 (**0.09**) | 77 (**0.19**) | 214 (**0.18**) | 3762 |
| *Saline* | 2649 (**0.07**) | 30 (**0.07**) | 85 (**0.07**) | 2764 |
| Total | 38045 | 402 | 1181 | 39628 |
| | 1988 | | | |
| *NoRisk* | 26959 (**0.85**) | 1144 (**0.21**) | 1434 (**0.60**) | 29537 |
| *PotSaline* | 2065 (**0.07**) | 1832 (**0.33**) | 413 (**0.17**) | 4310 |
| *Saline* | 2424 (**0.08**) | 2528 (**0.46**) | 559 (**0.23**) | 5511 |
| Total | 31448 | 5504 | 2406 | 39358 |
| | 1994 | | | |
| *NoRisk* | 24322 (**0.90**) | 1804 (**0.65**) | 3782 (**0.41**) | 29908 |
| *Saline* | 2806 (**0.10**) | 982 (**0.35**) | 5523 (**0.59**) | 9311 |
| Total | 27128 | 2786 | 9305 | 39219 |

Table 7.4: Mapping accuracies over validation data.

| Expert | Network | | | |
|---|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* | Total |
| | 1977 | | | |
| *NoRisk* | 30289 (**0.85**) | 797 (**0.76**) | 2016 (**0.70**) | 33102 |
| *PotSaline* | 3045 (**0.09**) | 179 (**0.17**) | 538 (**0.19**) | 3762 |
| *Saline* | 2380 (**0.06**) | 75 (**0.07**) | 309 (**0.11**) | 2764 |
| Total | 35714 | 1051 | 2863 | 39628 |
| | 1988 | | | |
| *NoRisk* | 24534 (**0.90**) | 1734 (**0.27**) | 3269 (**0.58**) | 29537 |
| *PotSaline* | 1320 (**0.05**) | 1986 (**0.31**) | 1004 (**0.18**) | 4310 |
| *Saline* | 1471 (**0.05**) | 2673 (**0.42**) | 1367 (**0.24**) | 5511 |
| Total | 27325 | 6393 | 5640 | 39358 |
| | 1994 | | | |
| *NoRisk* | 20451 (**0.93**) | 2431 (**0.77**) | 7026 (**0.50**) | 29908 |
| *Saline* | 1550 (**0.07**) | 721 (**0.23**) | 7040 (**0.50**) | 9311 |
| Total | 22001 | 3152 | 14066 | 39219 |

Table 7.5: Neighbourhood modified mapping accuracies over validation data.

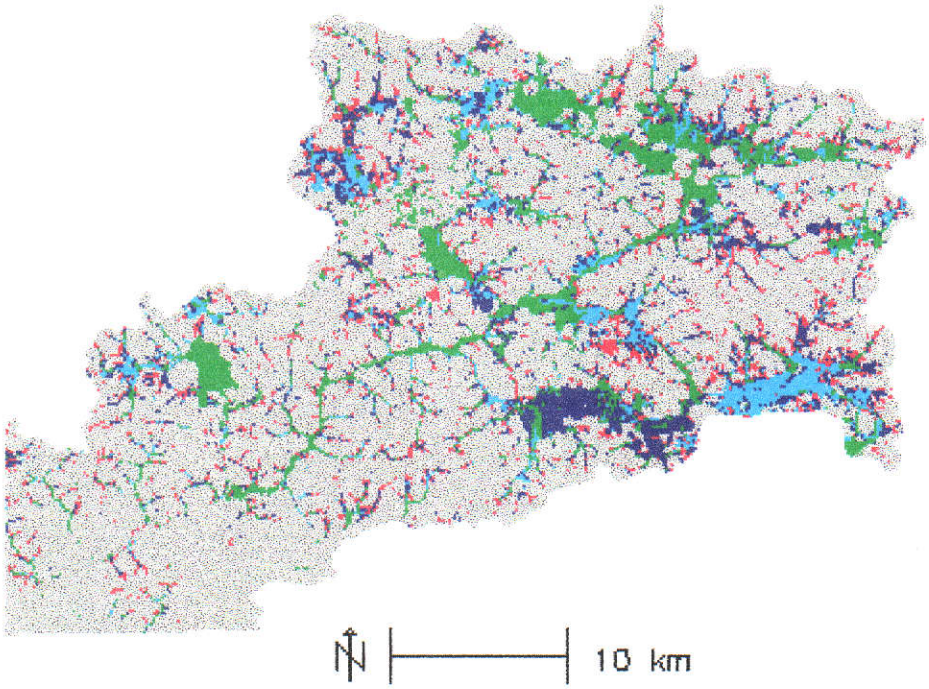



Figure 7.12: Catchment summary, neighbour modified results. The colours green, cyan, blue, red and grey identify land which was mapped as *Saline* in 1977, 1988, 1994, *PotSaline* in 1994 and *NoRisk* until at least 2004 respectively.

neighbourhood-modified outputs overestimate the extent of salinity and therefore provide an upper bound on the extent of salinisation as compared with the non-modified results which underestimate the extent and therefore provide a lower bound on the extent of salinisation — the *true* figures (8%, 13% and 22% salt affected for the years 1977, 1988 and 1994 respectively), as estimated by the expert from the training and validation areas, typically lies somewhere between these two figures. For instance, in 1977 the conservative estimate of salinity (table 7.6) is 5% of the catchment being affected, whereas the upper bound on the extent of salinity (table 7.7) is 9%. In the following this range will be represented as [5%,9%].

From the table, we observe that the percentage of land estimated to be affected by salinity increased from [5%,9%] in 1977, to [8%,14%] in 1988 and to [13%,20%] in 1994. This is expected to increase to [17%,27%] by the year 2004.

Of interest to the experts was the trend of salinity in cleared areas used for

| Year | Network | | |
|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* |
| 1977 % | 94 | 1 | 5 |
| 1988 % | 89 | 3 | 8 |
| 1994 % | 83 | 4 | 13 |

Table 7.6: Catchment summary. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*.

| Year | Network | | |
|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* |
| 1977 % | 89 | 2 | 9 |
| 1988 % | 82 | 4 | 14 |
| 1994 % | 73 | 7 | 20 |

Table 7.7: Catchment summary — neighbour-modified results. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*.

agriculture and in existing areas of remnant vegetation. These results are summarised in tables 7.8, 7.10, 7.9 and 7.11. The results suggest that [11%,15%] of total remnant vegetation was affected in 1977 and this increased to [13%,19%] in 1988. The amount of remnant vegetation identified in 1994 was less than 1988, but of the vegetation identified, roughly [13%,18%] was considered to be affected by salinity. The relatively high proportion of remnant vegetation affected in 1977 can partially be attributed to the fact that vegetation typically is left in low-lying areas, which are prone to salinisation and therefore may have degraded (died) during this period.

For the agricultural region, approximately [3%,7%] was mapped as *Saline* in 1977, [5%,11%] in 1988 and [13%,21%] in 1994. This is expected to increase to [18%,29%] by the year 2004 if no remedial action is taken.

| Year | Network | | |
|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* |
| 1977 % | 96 | 1 | 3 |
| 1988 % | 92 | 3 | 5 |
| 1994 % | 82 | 5 | 13 |

Table 7.8: Cleared land catchment summary. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*.

| Year | Network | | |
|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* |
| 1977 % | 88 | 1 | 11 |
| 1988 % | 82 | 5 | 13 |
| 1994 % | 85 | 2 | 13 |

Table 7.9: Remnant vegetation catchment summary. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*.

| Year | Network | | |
|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* |
| 1977 % | 90 | 3 | 7 |
| 1988 % | 85 | 4 | 11 |
| 1994 % | 71 | 8 | 21 |

Table 7.10: Cleared land catchment summary — neighbourhood-modified results. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*.

| Year | Network | | |
|---|---|---|---|
| | *NoRisk* | *PotSaline* | *Saline* |
| 1977 % | 84 | 1 | 15 |
| 1988 % | 76 | 5 | 19 |
| 1994 % | 80 | 2 | 18 |

Table 7.11: Remnant vegetation catchment summary — neighbourhood-modified. Percentage of the catchment having the labels *NoRisk*, *PotSaline* and *Saline*.

## 7.7 Summary

This chapter considered the problem of applying computer-based techniques for the purpose of mapping the historical extent and predicting the future extent of salinity in the Upper Kent River Catchment.

The methodology adopted in this chapter built upon the ideas presented in chapter 6: that is, the use of Landsat TM data for information on landuse, digital elevation data for information on landform, and the incorporation of other available data such as that provided by experts.

The application of the techniques developed in earlier chapters was modified slightly to better suit the emphasis on salinity prediction as opposed to land-condition monitoring and also the differing characteristics of the Upper Kent Catchment as compared to the MKG region in which most of the techniques were originally devised. For instance, the percentage of remnant vegetation preserved in the Kent Catchment is higher than that in the MKG region, as is the average annual rainfall as compared to the MKG region.

To be more specific, MLC of Landsat TM data (Evans, 1995b) provided information on historical landuse. Information on landform morphology was extracted from the DEM using the `ALLFLOW3` algorithm. The historical clearing information obtained from the classification of Landsat TM data was then used to initialise the `ALLFLOW3` algorithm to form estimates of the total and percentage of land upslope of a given location that had been cleared[2]. The slope of the land was extracted from the DEM and a surrogate variable for the expected ground water salinity was derived from the study of landform patterns by Ferdowsian (1994, 1993). These data were combined using a Bayesian network where the structure was derived from consideration of expert knowledge. The parameters of the network (rule weights) were mostly derived from data, although some human intervention was also used.

[2]The rationale for this step is that salinity is often attributed to clearing, that is, uncleared catchments typically exhibit minimal secondary salinity.

The application of the Bayesian network produced salinity risk assessments (raster images or *maps*) for the years 1977, 1988 and 1994. For each period, the salinity risk maps labelled a site as having no risk of salinity in the current time period, already affected by salinity or as being expected to go saline in the current time period. Noting that the output of the Bayesian network consistently underestimated the extent of salinity, a neighbourhood-modified version of the outputs was also produced. The neighbourhood-modified version essentially expanded the extent of the salinity by including regions immediately neighbouring those salt-affected areas produced by the unmodified results. Together, these two outputs provide a lower and upper bound on the extent of salinisation for each time period.

The accuracy of the salinity risk maps was then assessed by comparing them to ground truth. The total interpretation accuracies were 78%, 73% and 70% for the periods commencing in 1977, 1988 and 1994 respectively. Although these overall accuracies appear quite promising, some deficiencies in the accuracy for mapping particular classes existed. The class *NoRisk* dominates these figures and was mapped with an accuracy, *NoRisk* versus *NoRisk*, ranging from approximately 81% to 96%.

Closer analysis of the figures reveals that some care must be taken in drawing conclusions from the salinity risk maps produced by the network. For instance, in the validation areas the salinity risk maps derived from the network estimate the percentage of salt-affected land in these regions as between [3%,7%] in 1977, [6%,14%] in 1988 and [24%,36%] in 1994 as compared with the estimates by the expert of 7%, 11% and 24% respectively. That is the experts estimate falls within the bounds estimated from the network, apparently giving a degree of confidence in the extent of salinity estimated by the network for the catchment as a whole. However, the extent of the region only agrees with the expert some 60% of the time.

The accuracy of the salinity status of any specific site is not so good, with much confusion existing between the three classes. This is not surprising given the subtle

definition of salinity and the strict requirement that those areas considered to be potentially saline must be saline by the next time period.

As a result, questions of the form:

> Given that a site at some specified time is not affected by salinity, will it be salt-affected within the next time period?

cannot be accurately answered. However, the network results can give information relevant to the questions:

> Is this site more prone to salinity than others? Will planting/clearing trees within certain areas of this subcatchment positively/negatively affect this site?

which provides a rudimentary start to planning.

# Chapter 8

# Further Research

Research described in this thesis may roughly be classed as belonging to two fields of interest: (1) algorithms for deriving variables from DEMs; and (2) the use of Bayesian methods for combining data from remote sensing, GIS and expert sources. Experiments performed drew from research performed in both areas and applied the approaches developed to the real application of land-condition monitoring, with an emphasis on salinity mapping and prediction.

From a remote sensing/GIS perspective, the Bayesian approaches considered differ significantly from previous applications of the theory in that the evidence propagation scheme considered combines all data for a given pixel simultaneously[1] and provides updated joint marginal probabilities for the relevant groups of variables[2]. The approach forms the basis for current probabilistic expert systems and hence a link between traditional classification methods, data combination within a GIS environment and probabilistic expert systems is established. The application of this approach is a relatively new development in this domain and a number of research topics dealing with extensions to this approach are the basis of this chapter.

From an application perspective, deriving variables from DEMs proved in-

[1]Loosely speaking in expert system terminology this corresponds to performing both forward and backward inferences.

[2]This provides a method for reducing the effort required to extract rule weights from experts.

valuable to the results of salinity mapping and prediction. The usefulness of the digital elevation data as compared to other sources of ancillary data (for example soil types) is that complete coverge of the Western Australian wheatbelt is available from the Department of Land Administration.

For the applications considered here, the major limitation of the digital elevation data was the relatively large sampling intervals at which the data were originally collected, compared to the relief of the land. The sampling intervals vary from 5m (in elevation) in some regions to 20m in others. The impacts that the sampling interval have on the accuracy of the resulting land condition interpretations have been demonstrated in Chapter 5. Clearly, an important area for future research in this area is how to obtain higher-accuracy broad-scale DEMs at an affordable cost. Radar technologies or advances in automated elevation extraction from stereo sources of visible information (such as aerial photography/SPOT) may be worth future consideration, but are not discussed further here.

Considered here are two topics dealing with combining remotely sensed and GIS data using Bayesian networks.

The first topic considered briefly (section 8.1) outlines a potential means for incorporating methods of modelling data uncertainty with the data combination scheme used in this thesis.

Incorporation of data uncertainties and (iterative) neighbourhood operations significantly increase the computational requirements of the mapping process. As shown by Caccetta et al. (1995c), massively parallel processing offers a means to alleviate the computational burden. In section 8.2, a potential scheme for using massively parallel processing is explored.

## 8.1 Model Sensitivity to Data Uncertainty

### 8.1.1 Uncertainty in Data

Many types of data uncertainty may be defined (for example see (Goodchild and Gopal, 1989)), although here we will restrict ourselves to two types: attribute uncertainty; and positional uncertainty. In the context of this discusion, attribute uncertainty is associated with error in classifying data as belonging to a particular class. An obvious example is the MLC of Landsat TM data, whereby even if it is known (from other means) that a pixel has class $X$, say, the data and/or classification process may have errors in it resulting in some element of doubt in the resulting class label. Such classifiers typically output the most likely class label along with the posterior probability of the class label. For example, a class BareSalt may be the most likely attribute for a sample of data but its choice may not be definitive.

Figure 8.1 gives an example of a classification of Landsat TM data obtained from MLC and figure 8.2 the posterior probability for the class *BareSalt*. From the figures, it is observed that the class label is not identified with absolute certainty in many cases.

Positional uncertainty reflects the scenario that even if the *type* of feature can be identified (that is, its attribute/class can be specified), error may exist in the knowledge of the location of the object relative to other features of interest. In the context of the current discusion, an example of positional uncertainty is the imperfect registration of different map layers.

Positional uncertainty was considered, along with others, by Kiiveri (1996). As noted by the author, if we assume a rectangular array of pixels with expected registration errors taken to be equal in the $x$ and $y$ directions, then calculation of probabilities associated with positional uncertainty can be done by convolving the (binary) raster image for a particular class with a Gaussian filter. That is, if $g(x, y)$

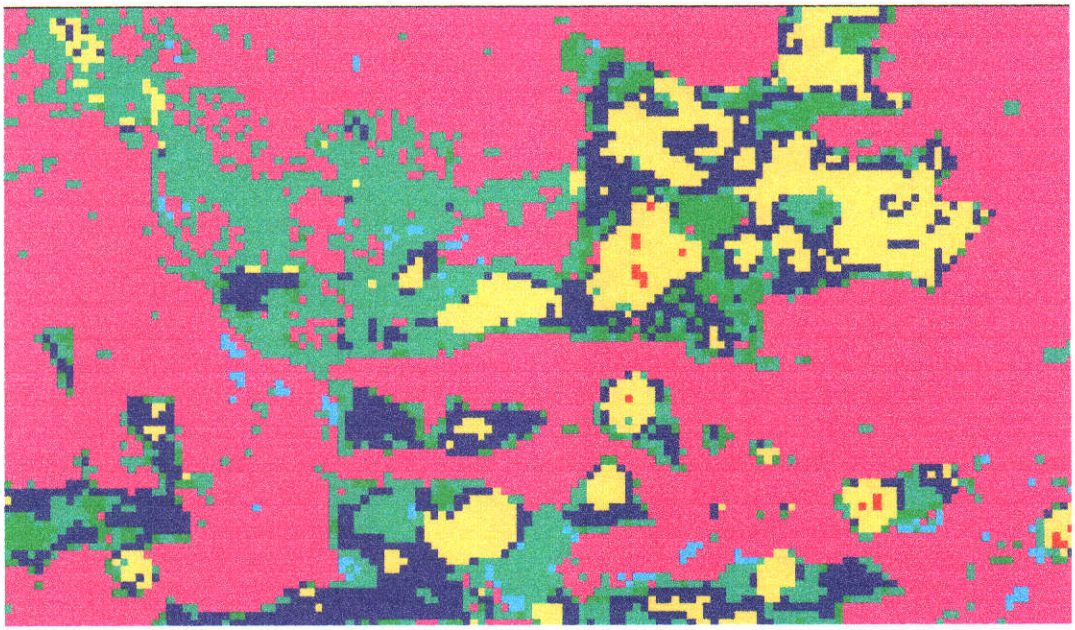


Figure 8.1: Landsat TM classification. Of interest here is the *BareSalt* class, which is coloured green. The other classes are *Water*, *RemVeg*, *BareSoil*, *WaterSideRemVeg*, *AgrLand* coloured red, blue, cyan, yellow and pink respectively.

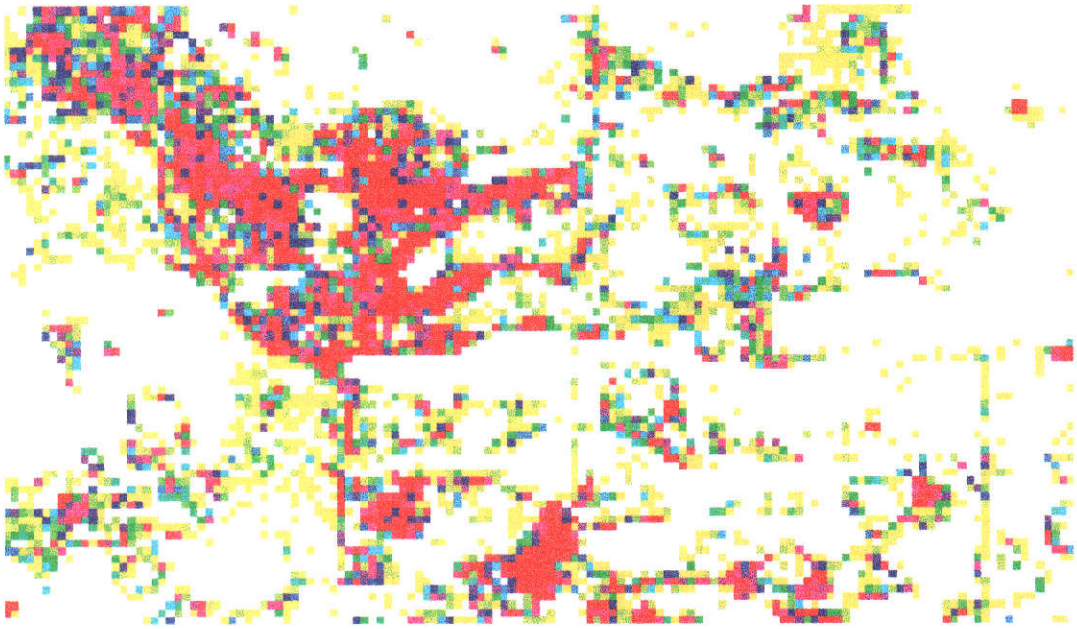


Figure 8.2: Attribute uncertainty in *BareSalt* class ($P(BareSalt)$). The figure is coloured red, which represents a high probability, through pink, blue, cyan, green and yellow, which represents a low probability. White represents $P(BareSalt) = 0$.

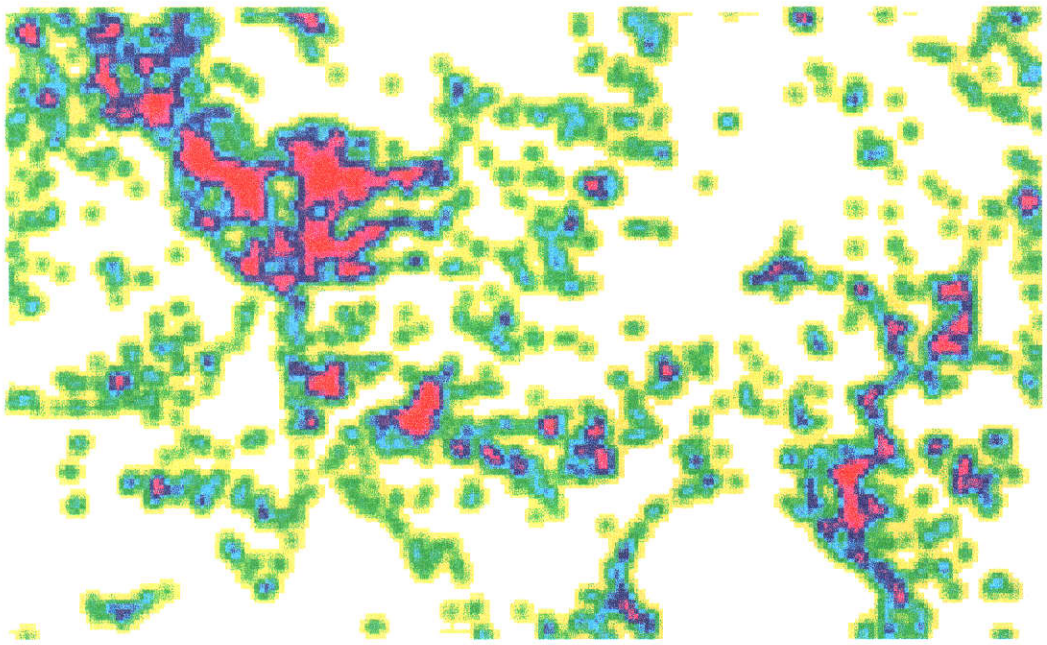


Figure 8.3: Probability of *BareSalt* derived from applying the Gaussian filter to the *BareSalt* class of figure 8.1. The figure is coloured red, which represents a high probability, through pink, blue, cyan, green and yellow, which represents a low probability. White represents $P(BareSalt) = 0$.

and $g_t(x_t, y_t)$ represent the images of the observed and true labels respectively, and

$$p(x_t = x, y_t = y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\{-\frac{(x_t-x)^2}{2\sigma_x^2} - \frac{(y_t-y)^2}{2\sigma_y^2}\}}$$

is the gaussian distortion function, then the probability $P(g_t(x_t, y_t) = g(x, y))$ is given by

$$P(g_t(x_t, y_t) = g(x, y)) = \int_x \int_y g(x, y) p(x_t = x, y_t = y) dy dx.$$

For example, consider the binary image constructed as follows: each pixel in the image has value 1 if it has attribute *BareSalt*, 0 otherwise. Assuming a Gaussian filter with $\sigma_x = \sigma_y = 1$ (pixel), the probabilistic image (figure 8.3) depicting the expected location of *BareSalt* results. For more details about distortion models for locational error, see Kiiveri (1996).

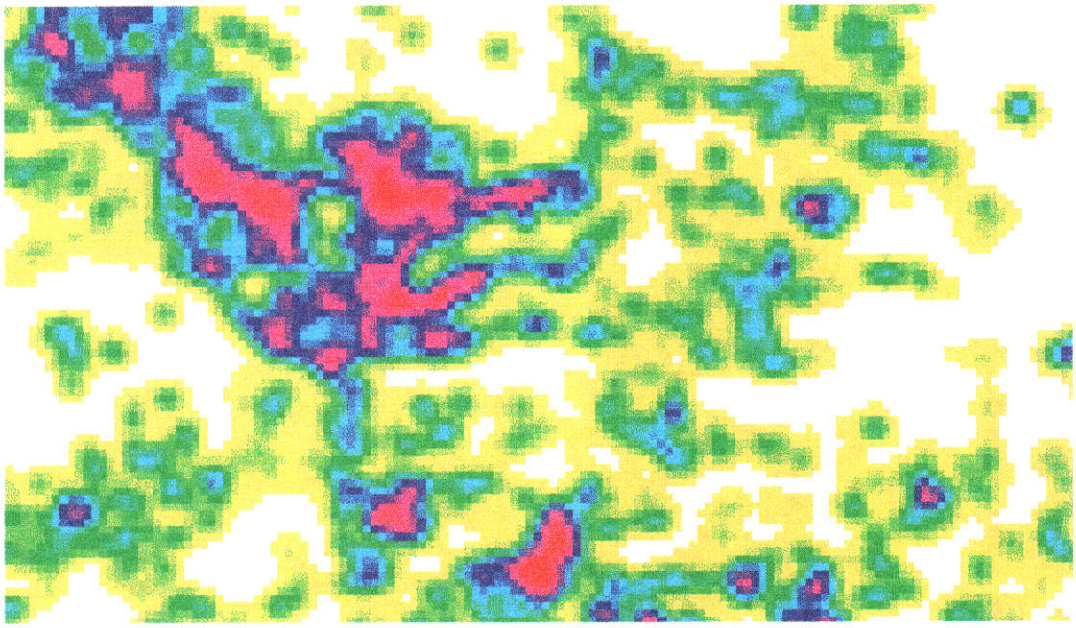


Figure 8.4: Probability of *BareSalt* including positional and attribute uncertainty. The figure is coloured red, which represents a high probability, through pink, blue, cyan, green and yellow, which represents a low probability. White represents $P(BareSalt) = 0$.

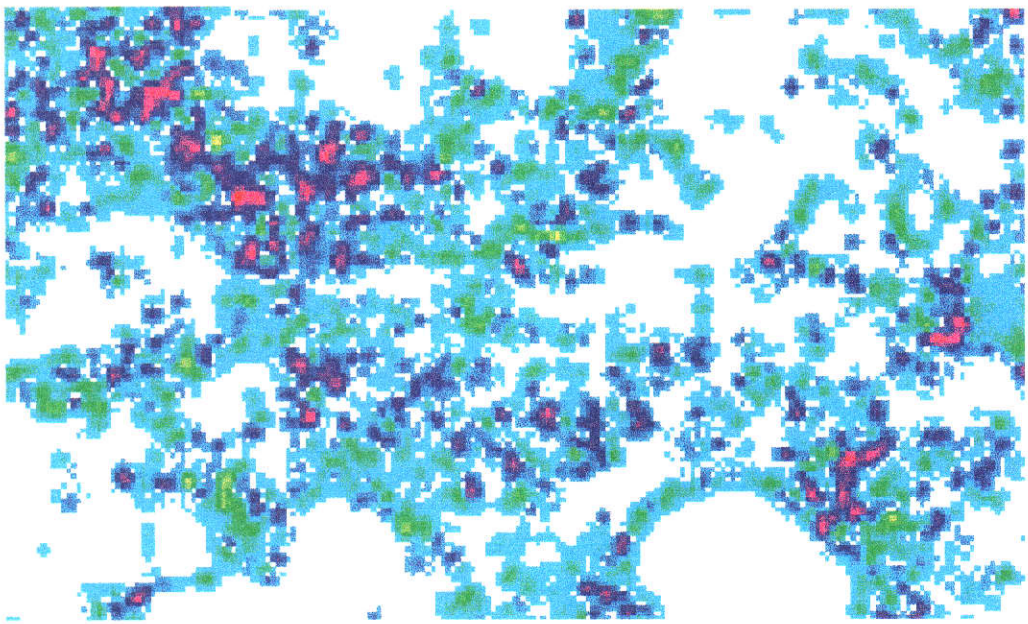


Figure 8.5: Difference of the images shown in figures 8.3 and 8.4. The figure is coloured red, representing a relatively large positive difference of approximately 0.25, through pink, blue, cyan, and green, which represents a relatively large negative difference of approximately 0.15. White represents no differences.

Figure 8.3 was constructed by assuming that no attribute uncertainty for the class *BareSalt* was present. From figure 8.2, it is known that this is not strictly true. In both instances, the resulting probabilistic images represent the expectation of observing a feature; in figure 8.2, uncertainty exists in the choice of label and in figure 8.3 the labels position is in doubt.

The following method is used to combine these approaches: for each class, the attribute uncertainty is derived and then convolved with the Gaussian filter. For example, figure 8.4 represents filtering the image of figure 8.2. For comparison purposes the difference of the two approaches, figures 8.3 and 8.4, is given in figure 8.5.

The latter approach has the following properties:

- Intuitively, the resulting image represents the expectation of observing a feature at a given location.
- The approach is computationally efficient.
- The resulting images can be readily incorporated in the data combination process.

A potential source of much attribute error is the stratification of continuous variables into a relatively small number of classes. For example, the variable PerUpClr94 used by the network of figure 7.9 was defined by stratifying the continuous variable defined in the range [0,100] into eight ordinal classes. For example, if $X$ represents a continuous variable, $Y = \{y_1, \ldots, y_k\}$ a set of $k$ discrete classes and $\delta$ a relatively small number, it would be expected (all other factors being equal) that a pixel with value $x - \delta$ would have similar properties to a pixel with value $x + \delta$. For example, if $k = 2$ and the classes $y_1$ and $y_2$ were defined as follows:

$$Y = \begin{cases} y_1 & \text{if } x < t \\ y_2 & \text{if } x \geq t \end{cases}$$

where $t$ is some (arbitrary) threshold, then this results in data with value $x - \delta$ and $x + \delta$ mapping to two different rules in the network. A way of reducing this effect would appear to be advantageous.

The approach considered here is to generate, for each continuous variable $X$, a probabilistic mapping from $X$ to the discrete variable $Y = \{y_0, \ldots, y_n\}$. That is, an image representing $P(Y = y_i)$, $i = 0$ to $n$, is generated. For example, given a set $A = \{a_0, a_1, \ldots, a_{n+1}\}$ of real numbers, one possibility is to generate $P(Y)$ as follows:

$$P(Y = y_i) = \begin{cases} 1 & \text{for } \frac{a_i + a_{i-1}}{2} < X \leq \frac{a_{i+1} + a_i}{2} \\ 0 & \text{otherwise} \end{cases}$$

which has the same effect as stratifying the variable $X$ into discrete classes. Another possibility for generating $P(Y)$ is:

$$P(Y = y_i) = \begin{cases} 1 - \frac{X - a_i}{a_{i+1} - a_i} & \text{for } a_i < X \leq a_{i+1} \\ \frac{X - a_{i-1}}{a_i - a_{i-1}} & \text{for } a_{i-1} < X \leq a_i \\ 0 & \text{otherwise} \end{cases} \tag{8.1}$$

which avoids the problem of stratification previously mentioned. This mapping is graphically depicted in figure 8.6. The effect of the operation is that classes are represented as probabilistic transition zones. This form of mapping is common in applications of fuzzy set theory (see for example Heuvelink and Burrough (1993); Leung and Leung (1993)) , for which many different types of functions for producing *fuzzy* class boundaries have been proposed.

An appealing property of the above transformation is that, for each pixel in the resulting probabilistic image, the expected value of the variable $Y$ is equal to the true value of the original continous variable. That is,

$$E(Y|X) = \sum_i a_i P(Y = a_i) = X,$$

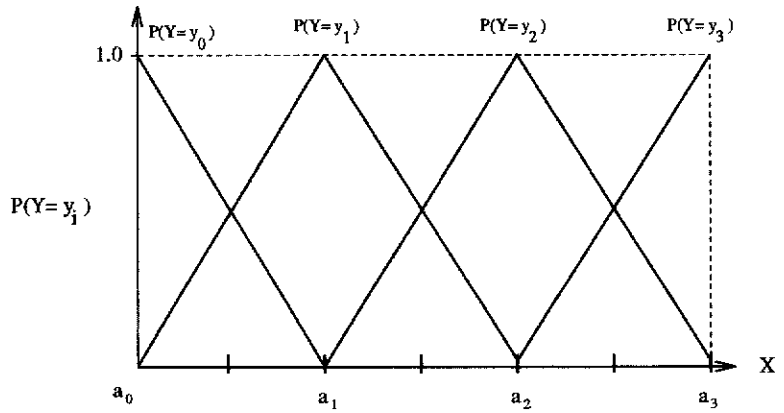



Figure 8.6: *Fuzzy* rule generation as defined in equation 8.1.

suggesting that the transformation is a better approximation to the original variable than that obtained from simple stratification.

With data attribute and positional uncertainties included, the model of figure 7.9 was applied to produce salinity risk maps over the validation areas. Comparison of the accuracy of the resulting classification with that obtained in Chapter 7 revealed that little improvement (<1%) was obtained by incorporating the data uncertainties, implying that the process used in Chapter 7 is fairly robust (ie not sensitive) to the types of data errors considered here. However, the results serve to illustrate the approach and may prove useful in other applications. Data obtained over one of the validation areas is presented below for illustration purposes. Here the changes that result in the underlying probability of an area being saline are focussed on. This illustrates how the incorporation of data uncertainties has an impact on the choice, and confidence associated with the choice, of the resulting class label.

Figure 8.7 depicts the ground truth (1994) for the region as mapped by the expert, where the green regions are considered to be affected, or potentially affected, by salinity and the red areas are considered to be not at risk from salinity.

Figure 8.8 depicts the salinity *risk* as produced by the network as applied in chapter 7. Here the probability of salinity $(1 - P(NoRisk))$ is shown, as compared with the resulting (most likely) class label as presented in chapter 7. The

figure is coloured yellow, depicting the areas with the highest probability of being/becoming saline, through to green, blue and then red which depicts the areas with the lowest probability of being/becoming saline. We note from this figure that the general locality of the highest risk areas as mapped by the network are in agreement with those identified by the expert. From figure 8.8, we observe that the network identifies some areas as being at greater risk than others. This form of interpretation is lost when merely considering the final classification labels.

Figure 8.9 depicts the network results when data uncertainties are included, where the results shown in this figure have the same interpretation as in figure 8.8. For this example, continous variables were mapped to *fuzzy* variables, as previously described, and all inputs were processed for positional uncertainty, as previously described.

Comparing figure 8.9 with figure 8.8, it is observed that incorporating data uncertainties changes the *risk* which the network assigns to a given location. For example, the black regions shown in figure 8.10 represent those areas where the probability of salinity has changed *significantly* (a change of 0.25 or more) as a result of incorporating data uncertainties. In the context of producing a classification, these changes have an impact on the choice of a class label and also on the confidence associated with the choice made.

## 8.2 Towards a Massively Parallel Implementation

The use of probabilistic input images and the inclusion of iterative neighbourhood techniques significantly increases the computational requirements of combining large volumes of data using Bayesian networks. Ways of reducing the time required to produce the maps is required.

Computational performance is directly related to the size of the network. Thus reducing the size of the network appears an obvious approach to reducing the com-

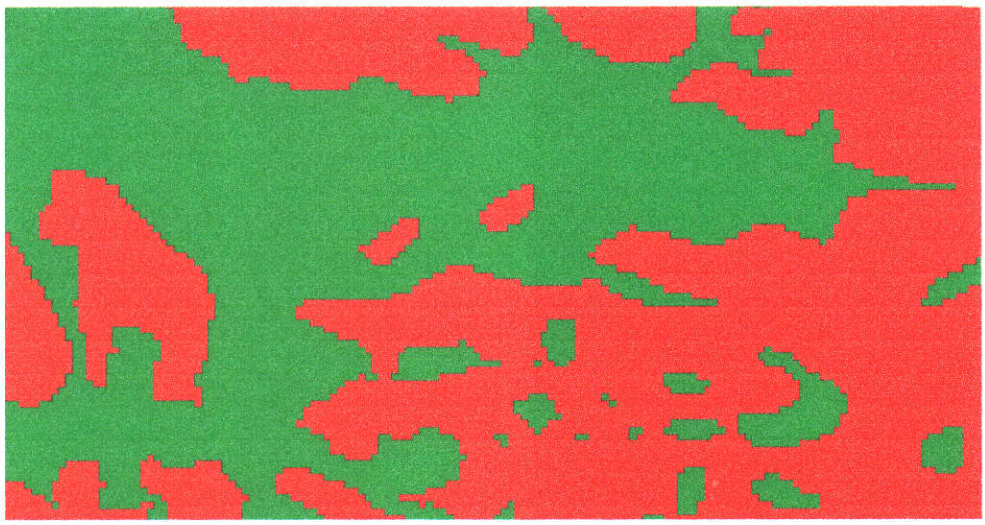


Figure 8.7: Salinity Ground Truth (1994). Saline (green), non-saline (red).

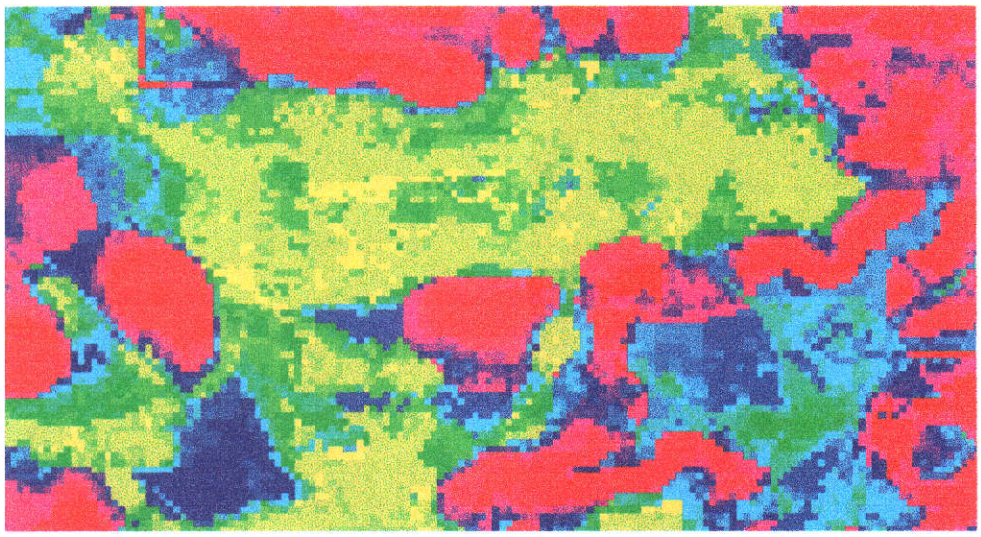


Figure 8.8: Network output — $P(NoRisk)$, (1994), without data uncertainties included. The figure is coloured red, representing a high (low) value for $P(NoRisk)$ ($P(SalinityRisk)$), through pink, blue, cyan, green and yellow, representing a low (high) value for $P(NoRisk)$ (($P(SalinityRisk)$).

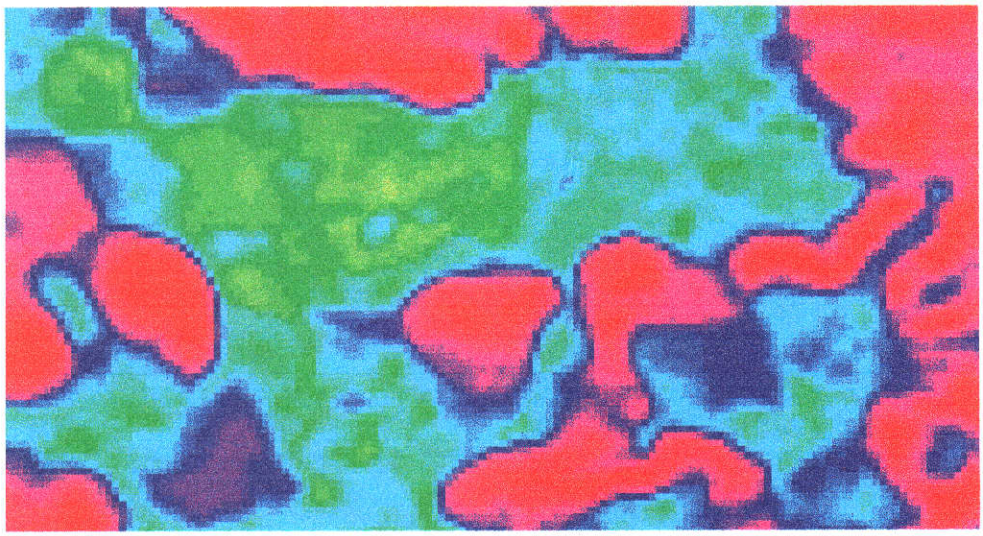


Figure 8.9: Network output — $P(NoRisk)$, (1994), with data uncertainties included. The figure is coloured red, representing a high (low) value for $P(NoRisk)$ ($P(SalinityRisk)$), through pink, blue, cyan, green and yellow, representing a low (high) value for $P(NoRisk)$ ($(P(SalinityRisk)$).

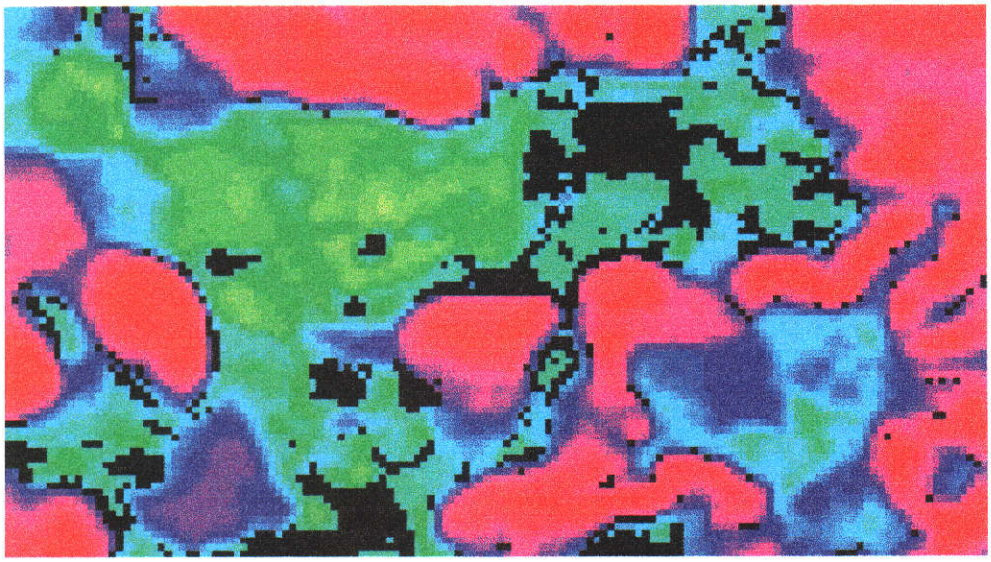


Figure 8.10: Network output — $P(NoRisk)$, (1994), where black represents areas which are *sensitive* to data uncertainty.

putation time. Although optimally triangulating a network produces a network of minimal size, the model may still be too large and need to be reduced by some other means.

One approach is to reduce the size of the network by removing edges that add little to the results. Such an approach has been considered by Kjaerulff (1993). In this approach, edges that contain weak (*statistical*) dependencies are removed from the model. Using this approach, edges may be removed until a computationally acceptable (*new*) model is derived, although achieving acceptable computational efficiency without removing important relationships may not always be possible.

Another possibility is to reduce the number of classes of selected variables, although this may not always be desirable. For example, landcover classes derived from MLC often have much thought in their formulation and would require considerable effort to redefine a smaller set. This option has the drawback that the network parameters would need to be redefined.

An approach that approximates a given network by removing parameters associated with unlikely events was considered by Jensen and Andersen (1990). Once the parameters have been removed, the parameter tables associated with each clique are *compressed* and computational improvements are achieved by making using of sparse techniques to access the parameters. This approach has the appeal that it does not change the structure of the network and hence the conceptual structure of the rules is the same, although results that are approximations to the original network are produced. A consequence of removing parameters is that outcomes associated with the removed parameters cannot be calculated, although the occurence of such cases is typically small, suggesting that much data may be processed with the compressed *efficient* model, and the undefined cases may be revisited with the original model.

While the above approaches reduce computational time by reducing the size of the network, another obvious way of reducing the required processing time is to make use of more powerful computers. Early work by Caccetta et al. (1995c)

showed that the use of massively parallel processing can reduce the time required for MLC by a factor of 20 as opposed to a SUN SPARC2 workstation. In this work, serial code of a commonly used MLC classifier was ported to a DECmpp SIMD machine. Such architectures are well-suited to image processing and have the advantage that more processors may be added. For example, in (Caccetta et al., 1995c), the machine was configured with 2K processors which is scalable to 64K processors, suggesting another factor of 32, some 600× faster in total, in the performance if the parallel machine is used as compared with the serial machine. Use of such architectures promise the most gain in processing time requirements.

With such an architecture, each processor can process data related to a single pixel. In the context of Bayesian networks, this equates to a Bayesian network per processing element. One limitation of the DEC MPP considered is that each processor has a limit of 64k of working memory which enforces a limit on the maximum size of a Bayesian network. Therefore, to make use of the computational advantages of the DEC MPP, it may be necessary to firstly reduce the size of the network. For example, the memory requirement of the network used in section 7.4 is approximately 190k, of which the majority of space is consumed by probability tables.

To examine the potential use of the DECmpp for processing data with Bayesian networks, the amount of compaction obtained from the scheme of Jensen and Andersen (1990), and the effect that this has on the results, is considered.

The parameter annihilation scheme of Jensen and Andersen (1990) was implemented and applied to the network of figure 7.9. Loosely speaking, the scheme works as follows (see Jensen and Andersen (1990) for more details):

- After `Initialisation()`, the normalized potentials associated with each clique have a (probability) mass equal to unity.
- The user then decides how much of the mass, typically expressed as a percentage, is to be retained.

- The scheme then proceeds as follows:

  For a given clique, identify the combination of variable states which has the lowest non-zero probability. Set this probability to zero. Proceed to the next lowest state and do likewise. Proceed until the sum of the probabilities removed equals the required percentage.

  For example, if the user requires 90% of the mass to be retained, then low state probabilities would successively be removed until their combined sum equalled 0.1.

- The new tables become the parameters for the network. Note here that sparse techniques are used to store only the states having a non-zero probability of occurrence. In this way, *compaction* is obtained by setting many states with small probabilities to zero.

Comment: The amount of compaction may be measured as the percentage of parameter combinations which have a zero probability of occurrence. This is the measure used here.

Various annihilation thresholds were chosen and the resulting compactions recorded in table 8.1. From the table, we observe that:

- Compaction increases as the amount of mass retained decreases.
- On average 21% of the original clique potentials had a zero probability of occurrence.
- A large degree of compaction can be obtained while still retaining a large proportion of the original clique mass. For example, specifying the retention of 90% of the mass results in a compaction of 82%.

Comments:

| Mass to be Retained (%) | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|
| Parameter Compaction (%) | 98 | 97 | 94 | 91 | 82 | 21 |

Table 8.1: Results of Network Parameter Compaction.

- This scheme obtains savings in storage space at the expense of approximating the original network parameters - the more compaction obtained, the more approximate are the resulting tables.
- Given the above point and the results presented in table 8.1, it would appear beneficial to retain at least 90% of the mass for the network considered.

Next, the effect of processing the data with the compacted parameters was examined by comparing the results with those obtained in chapter 7. A small subset of the region is given as an example in figure 8.8. The network parameters were compacted by specifying that 90% of the mass should be retained. The data were then processed with the new parameters. For the region considered in chapter 7, 76.2% of the region had results derived from the compacted parameters and the remaining 23.8% of the region was undefined by the new parameters. It is this 23.8% of the region which would have to be revisited with the original un-compacted model.

An example of the results is shown in figure 8.11, which may be compared with the results shown in figure 8.8. In figure 8.11, black represents those areas which could not be processed with the compacted parameters. These black regions represent those areas which would need to be revisited using the original parameters.

Comments:

- It is observed that the areas most at risk from salinity could not be processed with the compacted parameters, whereas those not at risk were generally represented by the modified parameters. This is not surprising when we consider that those areas not at risk have a temporally stable outcome as

opposed to the areas at risk which have a changing salinity status over the time period considered. In this way a temporally stable *NoRisk* class is highly probable, whereas in the regions where salinity is emerging, the temporal outcome is less clear and will generally be composed of many not-so-probable outcomes.

- Given the above point, application of a massively parallel implementation of the code to this problem would perform the following function:

  A parallel implementation would produce results for 76.2% of the catchment, effectively identifying those regions for which the (temporal) outcome is well defined.

  The remaining 23.8% of the region would be revisited with the original (serial) model and the the outcomes resolved.

- Assuming that the DEC MPP has limited RAM, but could possess 64k processors, then in theory the DEC MPP processing time would approach negligible amounts. In this way, processing time would be constrained by the proportion of area required to be processed with the serial machine.

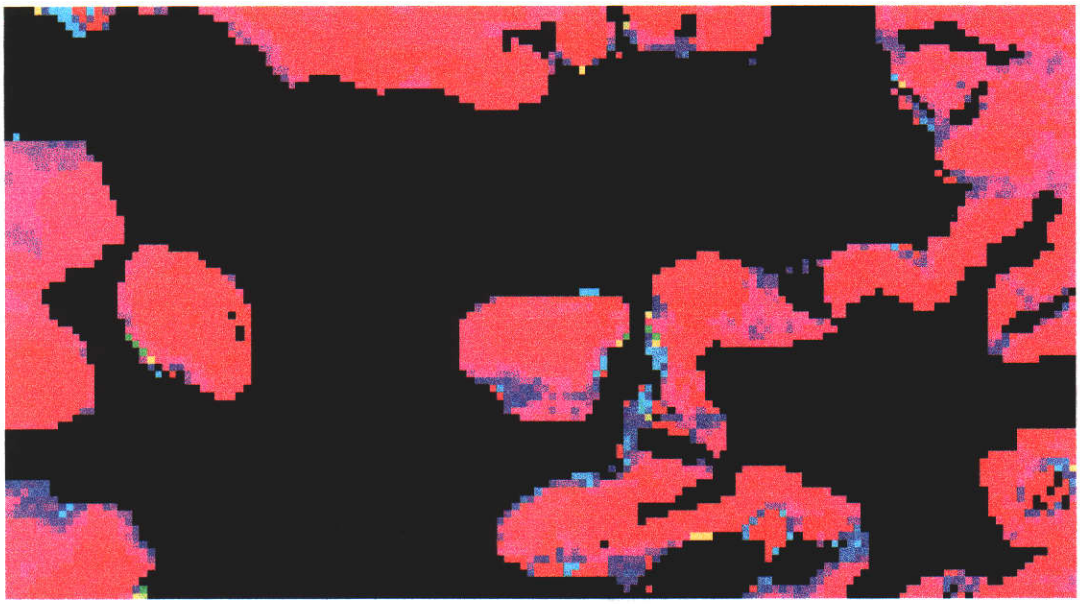


Figure 8.11: Results Using Compacted Parameters.

# Chapter 9

# Conclusions

The research described in this thesis primarily considers combining remote sensing, GIS and knowledge-based technologies for broad-scale mapping and monitoring of land condition in the West Australian agricultural region east of the Darling Scarp.

Previous work by other researchers in this domain has largely focused on the statistical analysis of remotely sensed Landsat TM data to provide broad-scale estimates of landuse and land condition. Typically, techniques such as maximum likelihood classification were used and the validity of using Landsat TM data for this purpose was established, although some limitations regarding the (spectral) confusion between certain landcovers remains.

This thesis considers methods which build upon the relative success of this previous research, with the primary goal being to improve upon the accuracies of broad-scale land-use/condition estimates.

The approach adopted is based on the notion that improvements in classification accuracy (and hence in broad-scale estimates) in this domain can be achieved by combining information derived from remotely sensed data with other forms of spatial data. The basis for this belief was derived from improvements reported by other researchers in the remote sensing literature.

Many different techniques for combining remotely sensed with other spatial data have been proposed in the remote sensing literature. A wider variety of tech-

niques for combining data/information existed in the Artificial Intelligence literature. Chapter 2.3 reviews some of the different approaches. Of the approaches, the use of expert system technology has appeal as it allows for the construction, maintenance, and relative ease of modifying potentially large sets of rules associated with combining multiple sources of data. Expert systems also have the appeal that they can be used interactively for decision support, answering what-if scenarios.

Of the different expert system/knowledge-based approaches for combining data, Bayesian networks have been adopted. An element of data error is typically associated with broad-scale data as the latter are often collected with cost constraints, or collected with other puposes in mind. Bayesian networks provide the basis of current probabilistic expert systems, allowing uncertainty in rule weights to be specified. This feature gives a mechanism for accounting for some of the data error. Other beneficial features of this approach include:

- All data are combined simultaneously and thus, unlike other systems, the resulting interpretation does not depend upon the order in which the data are presented.

- If for some reason some data are periodically missing, inferences can still be made from the remaining data.

- The maximum likelihood classifier is a special case of a Bayesian network, and therefore the approach is theoretically consistent with methods currently employed in the domain considered.

- Methods for estimating rule weights are well established in the statistical literature.

- Rules are expressed as conditional probability tables and are hence relatively easy to interpret.

- Resulting interpretations are expressed as probability distributions which give a the user a confidence rating of the resulting interpretation. Unlike other *fuzzy* approaches, if the rule weights have been derived from data, they may be interpreted as relative frequencies, which has an intuitive appeal.

One potentially important source of broad-scale information is digital elevation data, which are available from the Department of Land Administration for all areas of the West Australian agricultural region. The data were originally derived from the interpretation of stereo photography and consist of terrain locations sampled at fixed increments in elevation. The sampling increment varies for different regions. Typically the increment is either 5m, 10m or 20m.

Existing processing techniques and applications of this data are reviewed in section 2.5, and some potential limitations for the applications considered in this thesis are discussed. Experiments with existing techniques have been conducted in a region of the West Australian wheatbelt and the results are presented in Chapter 5. These experiments demonstrate that many of the existing techniques are inadequate for the monitoring tasks considered here.

Of most promise are techniques based on the prediction of flow paths. These techniques have been extended to produced a new flow path prediction algorithm which has been applied to the elevation data to provide a broad-scale partitioning of the terrain.

Using this new technique, the terrain has been partitioned into the (landform) classes: hilltops, ridges and upper-slopes, upper-valleys, lower valleys and broad valleys. The value of this partitioning procedure has been demonstrated by comparing the landform classes with ground truth information on salinisation. Using this data alone, it has been shown that for the area considered, 95% of all salinity was contained in the broad-valley class. For this class, approximately 73% of the area was salt-affected.

From a salinity mapping perpective, these initial results are encouraging, although it should be kept in mind that for the region considered, the broad-valley

class accounts for approximately 47% of the region, which is higher than that in the general landscape.

In Chapter 6, experiments using Bayesian networks for combining data were performed. These experiments focussed on the broader problem of obtaining better estimates of land-use/condition, for which salinity classes are a part. Experiments were based on an area of the wheat belt which receives fairly low average annual rainfall (approximately 300mm.) as compared to other areas. This region was chosen as it represents an area which traditionally has been difficult to map, as compared to higher rainfall regions, using MLC of Landsat TM data. The rationale was that if techniques could be developed in this region then they could easily be applied to other regions. Results from the experiments have been compared with MLC as its performance had been well documented by other researchers.

Experiments were conducted:

- To gain some insights into the likely effect of interpreting Landsat TM data with ground information.
- To gain some insights into the likely effects of combining Landsat TM data from successive growing seasons.
- To examine the effects of incorporating known data source error rates[1].
- To examine the effects of estimating model parameters (rule weights) using the EM algorithm as opposed to methods which are common in many current GIS.
- To examine the effects of incorporating spatial context into the classification process.
- To examine the role of prior knowledge in the derivation of rule weights.

[1]Traditionally part of the process of MLC is the evaluation of the classification error rates. Classifications from different years have differing accuracies and the models should in theory reflect these known differences.

Experiments with MODEL1 have demonstrated that interpretation accuracies for specific landcover classes can be improved by combining Landsat TM data with ground information. Further improvements have been achieved (MODEL2) by incorporating the classification error rates obtained from the MLC. Estimating the model parameters using the EM algorithm further improved the results of MODEL1 and MODEL2 as compared to a more obvious estimation technique.

While improving the accuracy of mapping salinity, MODEL1 and MODEL2 had an adverse effect on the mapping of paddock-use. The reason for this was that paddock-use was largely constrained by fence lines and cropping rotations more so than the many different soils found within each paddock. Further, for the time-scale considered, it was noted that the location of salinity should have been more temporally stable than was produced by the two models.

MODEL3 and MODEL4 were used to examine the potential role of temporal rules in landcover mapping. To isolate the effects that temporal rules would have, as opposed to ground data, ground data were ignored in these models. MODEL3 combined Landsat TM data from successive seasons. Loosely speaking, this model consists of a number of MLCs linked together with temporal rules. MODEL4 was the same as MODEL3 except, like MODEL2, it incorporated the classification error rates obtained from the MLC for each year.

MODEL3 and MODEL4 both consistently outperformed the MLC, confirming the usefulness of incorporating temporal rules. MODEL4 outperformed MODEL3, which demonstrated the advantage of incorporating data source error rates where possible. This was consistent with the results obtained from MODEL1 and MODEL2. Unlike MODEL1 and MODEL2, the temporal rules improved both salinity and paddock-use mapping accuracies.

The above models lacked spatial context, the basic idea being that a pixel is more likely to have a similar label to its immediate neighbours rather than a different label. In the domain of remote sensing, incorporating spatial context had previously been considered by other researchers, see for example (Kiiveri and

Campbell, 1992), although the approach was limited to the classification of a single source of remotely sensed data. In section 6.4.2, a new scheme for incorporating neighbourhood operations in a general Bayesian network has been proposed. The technique extends the approach of Besag (1986); Kiiveri and Campbell (1992).

To test the new scheme, neighbourhood operations were added to MODEL3 and MODEL4 to produce MODEL5 and MODEL6 respectively. These models consistently outperformed the MLC, the model of Besag (1986); Kiiveri and Campbell (1992), MODEL3 and MODEL4, demonstrating the usefulness of neighbourhood operations and the proposed new generalised scheme.

MODEL1, MODEL2, MODEL3, MODEL4, MODEL5 and MODEL6 all relied on having spectral estimates of the different landcover components and used this information to infer, via the EM algorithm, the rule weights for the other rules contained in each model. While prior knowledge was used to specify the structure of the networks, no prior knowledge concerning the rule weights was used.

MODEL6* was devised to examine the potential benefits of using prior knowledge in the specification of rule weights. This model differed from MODEL6 only in the way the rule weights were obtained: prior knowledge was used to initialise some rule weights before the application of the EM algorithm. MODEL6* outperformed MODEL6, demonstrating the benefits of having prior knowledge of the rule weights before commencement of the EM algorithm.

Finally in section 6.5, various components of the above models and the landform partitioning technique developed in Chapter 5 were brought together and a model which combines temporal Landsat TM, ground data and human expert knowedge was devised. The emphasis here was on reproducing salinity as mapped by an expert and the potential role of using a knowledge-based approach to summarise the relationships that exist between the many different sources of spatial data.

Human expert knowedge was used to define the structure of the network and initial estimates of the strength of the rules. The EM algorithm was then used to

estimate those parameters which were not specified and to refine those for which initial estimates were available.

In terms of salinity mapping, this approach reproduced, with a greater degree of accuracy than from previously considered models, the map as provided by a salinity mapping technician, although the definition of what the technician considered as saline differed somewhat from the definition as used in the earlier experiments[2].

In addition to combining data for the purpose of producing landcover interpretations, section 6.5 demonstrates the way in which Bayesian networks may be used to summarise the relationships obtained from the data and interactively examine *what-if* scenarios. This demonstration forms a link between using the approach to produce maps and the potential to form probabilistic decision support systems in the future.

Chapter 7 considers an application of the techniques developed to the problem of predicting salinity. The project was carried out in a region of the wheat-belt which differed significantly both in landform and landcover from the region in which the techniques were developed, providing some indication of the portability of the approach.

The basis of the project was to combine relevant data sets which were available to form historical estimates and the likely extent of future salinity for the Upper Kent River Catchment. The predictions should match those that would be mapped by a technician/expert.

The approach adopted in chapter 7 is based on the following:

- Historical Landsat TM data were used to provide information on the historical land-use.

- The landform partitioning techniques developed in chapter 5 were used to

[2]Earlier experiments were based on the notion that indicator species define saline from non-saline land. This definition is somewhat stricter than that used by the technician, who considered land to be saline if productivity was in someway affected by the presence of saline ground water.

form a partitioning of the terrain.

- An agricultural expert provided samples of the historical and expected future extent of salinity for a number of sites in the catchment. Knowledge gained from the expert was also used to form new variables appropriate to salinity prediction as opposed to salinity mapping. Derivation of these new variables was based on the landform partitioning techniques developed in section 5. Expert knowledge was also used to specify the structure of the network and the derivation of the network rule weights and finally to validate the accuracy of the mapping/predictions produced by the network.

For the validation areas, the historical salinity estimates produced by the network compared favourably with those produced by the expert. The network estimated the extent of salinity in the catchment to be in the range [3%,7%], [6%,14%] and [24%,36%] as compared to the experts estimate of 7%, 11% and 24% for the years 1977, 1983 and 1994 respectively.

Also noted in chapter 7 were some points of caution in the interpretation of the prediction results, the main point of caution being that while to a large degree the network correctly estimated the extent of salinity, confusion exists in the results as to when the onset of salinity (and therefore future salinty) actually occurred. This was seen to be due in part to the poor spectral discrimination of saline, potentially saline and non-saline sites obtained from Landsat TM in this region. To a large degree, predictions relied heavily on the landform varables derived from the DEM and therefore are partially bound by the errors in landform partitioning as decribed in section 5.

Finally, in section 8, some ideas for future research topics are outlined, in particular a scheme for incorporating data (spatial) uncertainty in the process of combining data and a scheme for improving the computational performance of probability propagation using Bayesian networks.

From an application point of view, future research in obtaining better salinity predictions could focus on reducing the errors in the landform partitioning

techniques, which have been discussed in detail in chapter 5.

# Appendix A

# Algorithm Pseudo Code

## A.1 Pseudo Code for HeightAbove().

```
HEIGHT_ABOVE(DEM,F)

/* DEM - raster digital elevation model   */
/* F   - binary image of features: 1 == feature, 0 == no feature */
/* D   - working array to hold distance image  */
/* P   - working array of points  */
/* L   - working array of unique pixel labels  */
/* H   - working array to hold height of closest feature pixel   */
/* HA  - working array to hold heigh above image  */

begin

/* Initialise distance image to zero/infinity  */
for all p=(i,j)
  begin
    if (F(p) > 0) then D(p) := 0
    else D(p) := infinity
  end
```

```
/* Uniquely label all feature pixels  */
counter := 1
for i = 1 to m  for j = 1 to n
  begin
    p = (i,j)
    if (F(p) > 0) then
      begin
        L(p)    := counter
        P(p)    := p
        counter := counter + 1
      end
  end

/* Propagate distances (and labels) from top */
/* left to bottom right  */
for i = 2 to m-1 for j = 2 to n-1
  begin
    p = (i,j)
    if ( (D(p) + d) < D(N) ) then
      begin
        D(N) := (D(p) + d)
        L(N) := L(p)
      end
  end

/* Propagate distances (and labels) from bottom */
/* right to top left  */
for i = m-1 to 2  for j = n-1 to 2
  begin
    p = (i,j)
    if ( (D(p) + d) < D(N) ) then
      begin
        D(N) := (D(p) + d)
        L(N) := L(p)
      end
```

```
  end

/* Form height of nearest feature pixel image  */
for i = 1 to counter-1  H(i) := DEM(P)

/* For height above image  */
for i = 1 to m  for j = 1 to n
  begin
    p = (i,j)
    HA(p) := DEM(p) - H(L(p))
  end

end
```

## A.2 Pseudo Code for INFLOWS().

```
INFLOWS(DEM,T)

/* DEM  - Raster digital elevation model */
/* p - pixel location */
/* N - 3X3 neighbourhood of p */
/* T - Slope threshold to define "flat" */
/* d - Pixel size (m): scale factor.  */
/* S - Raster image of slope "symbols" (+,-,=) */
/* R - Percent inflow variable (scalar) */
/* I - Output image "percentage inflow" */

start

/* For each pixel assign label +,-,= */
for i = 1 to n for j = 1 to m
  begin
    p = (i,j)
    max := 0
```

```
    for all n in N
      begin
        v := |(DEM(n)-DEM(p))| / d
        if (v > max)
          begin
            max := v
            if (DEM(n)-DEM(p) < 0) symbol := "-"
            else if (DEM(n)-DEM(p) > T) symbol := "+"
            else symbol := "="
          end
      end

    S(p) := symbol
  end

/* For each flat region, calculate the percentage  */
/* inflow and relabel the region with this value */
for i = 1 to n for j = 1 to m
  begin
   p = (i,j)
   if ( S(p) == "=") then
     begin
       R := FloodFill(S,p)
       FloodFill2(S,p,I,R)
     end
  end

  return(I)

finish
```

## A.3 Pseudo Code for ALLFLOW3().

```
ALLFLOW3(DEM,RAIN)
start

/* DEM  - raster digital elevation model */
/* RAIN - raster of ficticious rainfall event */
/* BVA  - raster holding flat area classification */
/* DMOD - DEM with pits filled in non-flat areas */
/* BV - image of broad valleys with modified  flow  */
/*   characteristics to ensure no flow feed-back */
/* ACCUM- resulting accumulation image */
/* N    - 3X3 neighbours of a given pixel p */

/**** INITIALISATION ****/
BVA  := INFLOWS(DEM,T)
DMOD := PIT_FILL(DEM,BVA)
BV   := ENSURE_CONSISTENT_FLOW_PATH(BVA,DMOD)

/* dump rain on landscape */
for i := 1 to n for j := 1 to m
  begin
    p     = (i,j)
    A(p)  := 0
    DELTA(p)  := RAIN(p)
  end

drained := N

/* while the landscape has not drained... */
while (drained == N)
  begin
    drained := N

    /* ...in non-flat regions pass all water to down */
    /* hill neighbouring pixels... */
```

```
  for i := 1 to n for j := 1 to m
    begin
      p := (i,j)
      if (( BV(p) == FALSE) and (DELTA(p) > 0)
        begin
          drained := Y
          total := 0
          for all n in N(p)
            begin
              v := DMOD(n)-DMOD(p)
              if (v < 0) total := total + v
            end
          for all n in N(p)
            begin
              v := DMOD(n)-DMOD(p)
              if (v < 0) then
                begin
                  A(n)     := A(n) + DELTA(p)/total
                  DELTA(n) := DELTA(n) + DELTA(p)/total
                end
            end
          DELTA(p) := 0
        end
    end

  /* ...and for all broad valley systems calculate the */
  /* total water flowing into region and pass all water*/
  /* to the regions outflow points  */
  for all b in BV
    begin
      R   := FLOOD_FILLA(b,DELTA)
      FLOOD_FILLB(b,DELTA,ACCUM)
    end
end
```

```
  return(ACCUM)

finish
```

# Bibliography


Adams, J. (1976). A probability model of medical reasoning and the MYCIN model. *Mathematical Biosciences*, 32:177–186.

Arnborg, S., Corneil, D., and Proskurowski, A. (1987). Complexity of finding embeddings in a $k$-tree. *SIAM Journal Discrete Mathematics*, 8:277–284.

Aspinall, R. (1992). An inductive modelling proceedure based on bayes' theorm for analysis of pattern in spatial data. *International Journal of Geographical Information Systems*, 6(2):105–121.

Badsberg, J. H. (1992). *Model search in contingency tables by CoCo*, pages 251–256. Neûchatel, Physica Verlag: Heidelberg.

Behn, G. and Campbell, N. A. (1992). Dieback assessment, using multispectral data, over the Stirling Range National Park, Western Australia. In *6th Australiasian remote sensing conference*, Wellington, New Zealand.

Besag, J. E. (1986). On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society Series B*, 48:259–302.

Bonham-Carter, G., Agterberg, F., and Wright, D. (1988). Integration of geological datasets for gold exploration in novia scotia. *Photogrammetric Engineering and Remote Sensing*, 54(11):1585–1592.

Borgefors, G. (1984). Distance transforms in arbitrary dimensions. *CVGIP*, 27:321–345.

Borgefors, G. (1986). Distance transforms in digital images. *CVGIP*, 34:344–371.

Bron, C. and Kerbosch, J. (1973). Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9).

Caccetta, P., Campbell, N., West, G., Kiiveri, H., and Gahegan, M. (1995a). Aspects of reasoning with uncertainty in an agricultural gis environment. *New Review of Applied Expert Systems*, 1:161–177.

Caccetta, P., Kiiveri, H., Evans, F., and Ferdowsian, R. (1995b). A knowledge-based approach to predicting salinity in the south-west of Western Australia. In *Proceedings of the 16th Asian conference on remote sensing, Korat, Thailand*, pages B–3–1 – B–3–6. Organizing Committee of World Tech'95, Thailand.

Caccetta, P. A., Campbell, N., and West, G. (1995c). A massively parallel implementation of an image classifier. In *16th Asian Conference on Remote Sensing, Korat, Thailand*, pages S–5–1 — S–5–6.

Campbell, N. and Wallace, J. (1989). Statistical methods for cover class mapping using remotely sensed data. In *Proceedings International Geoscience Remote Sensing Symposium*, pages 493–496.

Chickering, D. M., Geiger, D., and Heckerman, D. (1994). Learning Bayesian networks is NP-hard. Technical Report MSR-TR-94-17, Microsoft Corporation, Advanced Technology Division, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052.

Chorowicz, J., Ichoku, C., Riazanoff, S., Kim, Y., and Cervelle, B. (1992). A combined algorithm for automated drainage network extraction. *Water Resources Research*, 28(5):1293–1302.

Chorowicz, J., Kim, J., Manoussis, S., Rudant, J., Foin, P., and Veillet, I. (1989). A new technique for recognition of geological and geomorphological patterns in

digital terrain models. *International Journal of Remote Sensing of the Environment*, 29:229–239.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B:1–38.

Dempster, A. P. (1968). A generalization of Bayesian inference. *Journal Royal Statistical Society*, 30:205–247.

Evans, F. (1995a). Kent River study area - spectral analysis of 360 classified sites. Technical report, CSIRO Division of Mathematics and Statistics, Perth, Western Australia.

Evans, F. (1995b). Kent River study area, Landsat TM classification analyses. Technical report, CSIRO Division of Mathematics and Statistics, Perth, Western Australia.

Evans, F., Caccetta, P., Ferdowsian, R., Kiiveri, H., and Campbell, N. (1995). Predicting salinity, a report from the LWRRDC funded project *Integrating Remotely Sensed Data With Other Spatial Data to Predict Areas at Risk from Salinity*. Technical report, CSIRO Division of Mathematics and Statistics, W.A. and collaborating partners.

Ferdowsian, R. (1993). Landform patterns of western forest area, on the south coast of W.A. Technical report, Department of Agriculture, Western Australia.

Ferdowsian, R. (1994). Salinity study of the Kent River catchment. Technical report, Department of Agriculture, Albany, Western Australia.

Ferdowsian, R. and Greenham, K. (1992). Integrated catchment management: upper Denmark catchment. Technical Report 130, Department of Agriculture, Western Australia, Albany, Western Australia.

Flack, J. C. (1995). The role of process networks in binding remotely sensed data to a gis. In *Proceedings of the 2nd Australasian remote sensing and GIS Forum*, pages 159–164, Darwin, Northern Territory, AUSTRALIA.

Furby, S. (1994). Moora–Kalannie Goodlands study area: Class label vs reference site cover comparisons. Technical report, CSIRO Division of Mathematics and Statistics, Perth, Western Australia.

Furby, S., Wallace, J., Caccetta, P., and Wheaton, G. (1995). Detecting and monitoring salt-affected land. Technical report, CSIRO Division of Mathematics and Statistics, Leeuwin Center, Perth, Western Australia.

Garvey, T. D. (1987). Evidential reasoning for geographic evaluation for helicopter route planning. *IEEE Transactions on Geoscience and Remote Sensing*, GE-25(3):294–304.

Goodchild, M. F. and Gopal, S. (1989). *The Accuracy of Spatial Databases*. Taylor and Francis, London, UK.

Gordon, J. and Shortliffe, E. (1985). A method for managing evidential reasoning in a hierarchical hypothesis space. *Artificial Intelligence*, 26:323–357.

Graff, L. H. and Usery, E. (1993). Automated classification of generic terrain features in digital elevation models. *Photogrammetric Engineering and Remote Sensing*, 59(9):1409–1417.

Haralick, R. M. (1983). Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing*, 22:28–38.

Haralick, R. M., Wang, S., Shapiro, L. G., and Campbell, J. (1985). Extraction of drainage networks by using the consistent labeling technique. *Remote Sensing of Environment*, 18:163–175.

Heckerman, D. (1995a). A tutorial on learning Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Corporation, Advanced Technology Division, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052.

Heckerman, D. (1995b). A Bayesian approach to learning causal networks. Technical Report MSR-TR-95-04, Microsoft Corporation, Advanced Technology Division, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052.

Heckerman, D., Geiger, D., and Chickering, D. M. (1994). Learning bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Corporation, Advanced Technology Division, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052.

Henkind, S. J. and Harrison, M. C. (1989). An analysis of four uncertainty calculi. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(5).

Heuvelink, G. B. and Burrough, P. A. (1993). Error propagation in cartographic modelling using Boolean logic and continuous classification. *International Journal of Geographical Information Systems*, 7(3):231–246.

Hojsgaard, S., Caccetta, P., and Kiiveri, H. (1997). Pixel allocation using remotely-sensed data and ground data. *International Journal of Remote Sensing*, 18(2):417–433.

Hojsgaard, S. and Thiesson, B. (1995). BIFROST - block recursive models induced from relevent knowledge, observations, and statistical techniques. *Computational Statistics and Data Analysis*, 19:155–175.

Hutchinson, C. (1982). Techniques for combining Landsat and ancillary data for digital classification improvement. *Photogrammetric Engineering and Remote Sensing*, 48:123–130.

Janssen, L. and Middelkoop, H. (1992). Knowledge–based crop classification of a Landsat thematic mapper image. *International Journal of Remote Sensing*, 13:2827–2837.

Jensen, F. and Andersen, S. K. (1990). Approximations in Bayesian belief universes for knowledge-based systems. In *Proceedings of the sixth conference on uncertainty in artificial intelligence*, pages 162–169, Cambridge, Massachusetts.

Jensen, F. V. (1993). Introduction to bayesian networks. Technical Report IR 93-2003, Institute for Electronic Systems, Department of Mathematics and Computer Science, Aalborg University, Fredrik Bajers Vej 7E, DK 9220 Aalborg Ø, Denmark.

Jenson, S. K. and Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, 54(11):1593–1600.

Jordan, M. I. and Jacobs, R. A. (1993). Hierarchical mixtures of experts and the EM algorithm. Technical Report 1440, Massachusetts Institute of Technology Artificial Intelligence Laboratory and Center for Biological and Computational Learning Department of Brain and Cognitive Sciences.

Karnieli, A. M., Diskin, M. H., and LAne, L. J. (1994). Celmod5 - a semi-distributed cell model for conversion of rainfall into runoff in semi-arid watersheds. *Journal of Hydrology*, 157:61–85.

Kennett-Smith, A., Narayan, K., and Walker, G. (1994). The impact of change in agricultural land management on the groundwater and salt balances in a regional discharge zone. In *Water Down Under '94*, Adelaide.

Kiiveri, H. (1996). Assessing, representing and transmitting positional uncertainty in maps. *International Journal Geographical Information Science*, 11(1):33–52.

Kiiveri, H. T. and Campbell, N. A. (1992). Allocation of remotely sensed data using Markov models for image data and pixel labels. *Australian Journal of Statistics*, 34(3):361–374.

Kirk, D. (1992). pages 278–284. Academic Press Inc.

Kjaerulff, U. (1990). Triangulation of graphs - algorithms giving small total state space. Technical report, Aalborg University, Fredrik Bajers Vej 7E, DK 9220 Aalborg Ø, Denmark.

Kjaerulff, U. (1993). Approximation of bayesian networks through edge removals. Technical Report IR-93-2007, Institute for Electronic Systems, Department of Mathematics and Computer Science, Aalborg University, Fredrik Bajers Vej 7E, DK 9220 Aalborg Ø, Denmark.

Kontoes, C., Wilkinson, G. G., Burrill, A., Goffredo, S., and Mégier, J. (1993). An experimental system for the integration of gis data in knowledge-based image analysis for remote sensing of agriculture. *International Journal Geographical Information Systems*, 7(3):247–262.

Kruse, F. A., Lefkoff, A. B., and Dietz, J. B. (1993). Expert system-based mineral mapping in Northern Death Valley, California/Nevada, using the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing Environment*, 44:309–336.

Kweon, I. S. and Kanade, T. (1994). Extracting topographic terrain features from elevation maps. *CVGIP:Image Understanding*, 59(2):171–182.

Lauritzen, S. L. and Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50(2):157–224.

Lawler, L., Lenstra, A., Rinnooy, K., and Shmoys, D. (1985). *The travelling salesman problem.* John Wiley and Sons.

Lee, C.-H. (1988). A comparison of two evidential reasoning schemes. *Artificial Intelligence*, 35:127–134.

Lee, T., Richards, J. A., and Swain, P. H. (1987). Probabilistic and evidential approaches for multisource data analysis. *IEEE Transactions on Geoscience and Remote Sensing*, GE-25(3):283–293.

Leung, Y. and Leung, K. S. (1993). An intelligent expert system shell for knowledge-based geographical information systems: 2. some applications. *International Journal Geographical Information Systems*, 7(3):201–213.

Malcolm, C. (1993). Fodder shrubs for salt-affected land in South Australia. Technical Report 213, Primary Industries, South Australia.

Moon, W. M. (1990). Integration of geophysical and geological data using evidential belief function. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4):711–720.

Ngwenyama, O. K. and Bryson, N. (1992). A formal method for analyzing and integrating the rule-sets of multiple experts. *Information Systems*, 17(1):1–16.

Nulsen, R. A. (1981). Salt-affected land in the shire of Wongan-Ballidu, Western Australia. *Australian Journal of Soil Research*, 19:87–91.

Olesen, K. G. (1993). Causal probabilistic networks with both discrete and continuous variables. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(15).

Osherson, D., Shafir, E., and Smith, E. (1994). Extracting the coherent core of human probability judgement: a research program for cognitive psychology. *Cognition*, 50:299–313.

Palmer, M. J., Furby, S. L., and Wallace, J. (1994). Mapping wind erosion in the South Stirlings, a report from the LWRRDC project: Detecting and monitoring

changes in land condition through time using remotely sensed data. Technical report, CSIRO Division of Mathematics and Statistics, Perth.

Papoulis, A. (1985). *Probability, Random Variables, and Stochastic Processes*, pages 27–31. McGraw–Hill International Book Company, second edition.

Pearl, J. (1986a). Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29:241–288.

Pearl, J. (1986b). On evidential reasoning in a heirarchy of hypotheses. *Artificial Intelligence*, 28:9–15.

Qian, J., Enrich, R. W., and Campbell, J. B. (1990). DNESYS - an expert system for automatic extraction of drainage networks from digital elevation data. *IEEE Transactions on Geoscience and Remote Sensing*, 28(1):29–45.

Quinn, P., Beven, K., Chevallier, P., and Planchon, O. (1991). The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. *Hydrological Processes*, 5:59–79.

Rao, C. R. (1966). *Linear Statistical Inference and its Applications*, pages 289–291. John Wiley and Sons, Incorporated, New York, second edition.

Riazanoff, S., Cervelle, B., and Chorowicz, J. (1988). Ridge and valley line extraction from digital terrain models. *International Journal of Remote Sensing*, 9(6):1175–1183.

Schultz, G. A. (1994). Meso-scale modelling of runoff and water balances using remote sensing and other gis data. *Hydrological Sciences - Journal des Sciences Hydrologiques*, 39(2).

Shafer, G. (1976). *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, New Jersy.

Shenoy, P. P. and Shafer, G. (1986). Propagating belief functions using local computations. *IEEE Expert*, 1(3):43–52.

Shenoy, P. P. and Shafer, G. (1990). Axioms for probability and belief-function propagation. In *Uncertainty in Artificial Intelligence*. Elsevier science publishers.

Shortliffe, E. (1976). *Computer Based Medical Consultations: MYCIN*. Elsevier, New York.

Shortliffe, E. and Buchanan, B. (1975). A model for inexact reasoning in medicine. *Mathematical Biosciences*, 23:351–379.

Skidmore, A. K. (1989). An expert system classifies Eucalypt forest types using Thematic Mapper data and a digital terrain model. *Photogrammetric Engineering and Remote Sensing*, 55(10):1449–1464.

Strahler, A. (1980). The use of prior probabilities in maximum likelihood classification of remotely sensed data. *Remote Sensing of Environment*, 10:135–163.

Strahler, A., Logan, T., and Woodcock, C. (1979). Forest classification and inventory system using landsat, digital terrain, and ground sample data. *13th International Symposium on Remote Sensing of Environment*, pages 1541–1557.

Sucar, L. E. and Gillies, D. F. (1994). Probabilistic reasoning in high-level vision. *Image and Vision Computing*, 12(1):42–60.

Tarjan, R. and Yannakakis, M. (1984). Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduce acyclic hypergraphs. *SIAM Journal of Computing*, 13(3).

Ton, J., Sticklen, J., and Jain, A. K. (1991). Knowledge-based segmentation of Landsat images. *IEEE Transactions on Geoscience and Remote Sensing*, 29(2):222–232.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6).

Wallace, J., Campbell, N., Wheaton, G., and McFarlane, D. (1993). Spectral discrimination and mapping of waterlogged cereal crops in Western Australia. *International Journal of Remote Sensing*, 14(14):2731–2743.

Wallace, J. F. and Wheaton, G. A. (1990). Spectral discrimination and mapping of land degradation in Western Australia's agricultural region. In *5th Australasian Remote Sensing Conference, Perth, Western Australia*, pages 1066–1073.

Wang, F. (1990). Fuzzy supervised classification of remote sensing images. *IEEE Transaction on Geoscience and Remote Sensing*, 28(2):194–201.

Wang, F. and Newkirk, R. (1988). A knowledge-based system for highway network extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 26(5):525–531.

Wharton, S. W. (1987). A spectral-knowledge-based approach for urban land-cover discrimination. *IEEE Transactions on Geoscience and Remote Sensing*, GE-25(3):272–282.

Wheaton, G., Wallace, J., McFarlane, D., and Campbell, N. (1992). Mapping salt-affected land in Western Australia. In *Proceedings of the 6th Australasian Remote Sensing Conference*, volume 2, pages 369–377, Wellington, New Zealand.

Wheaton, G., Wallace, J., McFarlane, D., Campbell, N., and Caccetta, P. (1994). Mapping and monitoring salt–affected land in Western Australia. In *Proceedings of Resource Technology '94 Conference*, pages 531–543, Melbourne, Australia.

Wilkinson, G., Folving, S., Fullerton, K., and Mégier, J. (1992). A feasibility study on the automatic updating of the corine land cover database using satellite

imagery in portugal. In *Third European Conference on Geographical Information Systems*, Munich.

Wilkinson, G. G. and Mégier, J. (1990). Evidential reasoning in a pixel classification hierarchy-a potential method for integrating image classifiers and expert system rules based on geographic context. *International Journal Remote Sensing*, 11(10):1963–1968.