COPYRIGHT AND USE OF THIS THESIS

This thesis must be used in accordance with the provisions of the Copyright Act 1968.

Reproduction of material protected by copyright may be an infringement of copyright and copyright owners may be entitled to take legal action against persons who infringe their copyright.

Section 51 (2) of the Copyright Act permits an authorized officer of a university library or archives to provide a copy (by communication or otherwise) of an unpublished thesis kept in the library or archives, to a person who satisfies the authorized officer that he or she requires the reproduction for the purposes of research or study.

The Copyright Act grants the creator of a work a number of moral rights, specifically the right of attribution, the right against false attribution and the right of integrity.

You may infringe the author’s moral rights if you:

- fail to acknowledge the author of this thesis if you quote sections from the work
- attribute this thesis to another author
- subject this thesis to derogatory treatment which may prejudice the author’s reputation

For further information contact the University’s Director of Copyright Services

sydney.edu.au/copyright
Hierarchical Classification of Scientific Taxonomies with Autonomous Underwater Vehicles

Michael Bewley

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

March 2016
Supervisor
Prof. Stefan B. Williams

Associate Supervisor
Dr. Oscar Pizarro
For William, for inspiring me to keep going, Samuel, for inspiring me to finish, and Fliss, for being with me every step of the way.
# Contents

Abstract 3

Acknowledgments 5

1 Introduction 9
   1.1 Overview ................................................. 9
   1.2 Problem Statement ...................................... 10
   1.3 Contributions ......................................... 11
   1.4 Outline ................................................. 13

2 Single species classification on a large data set 17
   2.1 Introduction ............................................. 18
   2.2 Related Work ............................................. 20
   2.3 Methodology .............................................. 23
      2.3.1 Scale and Keypoints ................................. 23
      2.3.2 Descriptors ......................................... 24
      2.3.3 Classification ....................................... 26
   2.4 Data Set ................................................ 27
      2.4.1 Classes and Labelling ............................... 27
      2.4.2 Training, Validation and Test Sets ............... 28
      2.4.3 Dive Characteristics ................................ 30
   2.5 Model Comparison Experiments ......................... 31
      2.5.1 Patch, Feature and Classifier Combinations ....... 31
      2.5.2 Performance Evaluation .............................. 32
      2.5.3 Results ............................................... 32
      2.5.4 Discussion ........................................... 33
   2.6 Final Model Evaluation .................................. 34
      2.6.1 Training Set Sensitivity Tests .................... 35
      2.6.2 Error Analysis Tests ................................ 35
      2.6.3 Results and Discussion ............................. 36
   2.7 Conclusion ................................................. 39

3 Hierarchical Classification in AUV Imagery 45
   3.1 Introduction .............................................. 47
   3.2 Hierarchical Classification ............................ 49
      3.2.1 Classification ......................................... 49
      3.2.2 Training policies .................................... 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.3 Prediction</td>
<td>52</td>
</tr>
<tr>
<td>3.2.4 Performance measure</td>
<td>52</td>
</tr>
<tr>
<td>3.2.5 Descriptors</td>
<td>54</td>
</tr>
<tr>
<td>3.3 Data set and Hierarchy</td>
<td>55</td>
</tr>
<tr>
<td>3.4 Results and Discussion</td>
<td>57</td>
</tr>
<tr>
<td>3.4.1 Hierarchical Classification Approaches</td>
<td>57</td>
</tr>
<tr>
<td>3.4.2 Hierarchical vs flat multi-class classification</td>
<td>59</td>
</tr>
<tr>
<td>3.5 Conclusion</td>
<td>61</td>
</tr>
<tr>
<td>4 BENTHOZ-2015: An Australian benthic data set</td>
<td>63</td>
</tr>
<tr>
<td>4.1 Background &amp; Summary</td>
<td>65</td>
</tr>
<tr>
<td>4.2 Methods</td>
<td>67</td>
</tr>
<tr>
<td>4.2.1 AUV Data Collection</td>
<td>67</td>
</tr>
<tr>
<td>4.2.2 Expert Annotations</td>
<td>69</td>
</tr>
<tr>
<td>4.2.3 Code availability</td>
<td>73</td>
</tr>
<tr>
<td>4.3 Data Records</td>
<td>73</td>
</tr>
<tr>
<td>4.3.1 Expert Labels</td>
<td>74</td>
</tr>
<tr>
<td>4.3.2 Image Metadata</td>
<td>76</td>
</tr>
<tr>
<td>4.3.3 Images</td>
<td>77</td>
</tr>
<tr>
<td>4.3.4 Data set definitions</td>
<td>77</td>
</tr>
<tr>
<td>4.4 Technical Validation</td>
<td>78</td>
</tr>
<tr>
<td>4.4.1 Western Australia</td>
<td>79</td>
</tr>
<tr>
<td>4.4.2 New South Wales</td>
<td>79</td>
</tr>
<tr>
<td>4.4.3 Tasmania</td>
<td>80</td>
</tr>
<tr>
<td>4.4.4 Queensland</td>
<td>80</td>
</tr>
<tr>
<td>4.4.5 Geolocation Accuracy</td>
<td>81</td>
</tr>
<tr>
<td>4.5 Usage Notes</td>
<td>81</td>
</tr>
<tr>
<td>4.5.1 Automated Labeling Research</td>
<td>81</td>
</tr>
<tr>
<td>4.5.2 Marine Science</td>
<td>83</td>
</tr>
<tr>
<td>4.6 Acknowledgments</td>
<td>84</td>
</tr>
<tr>
<td>4.6.1 Author Contributions</td>
<td>84</td>
</tr>
<tr>
<td>4.6.2 Other Acknowledgments</td>
<td>85</td>
</tr>
<tr>
<td>4.7 Competing financial interests</td>
<td>86</td>
</tr>
<tr>
<td>4.8 Data Citations</td>
<td>86</td>
</tr>
<tr>
<td>5 Hierarchical Classification with Bayesian Networks</td>
<td>91</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>93</td>
</tr>
<tr>
<td>5.2 Background</td>
<td>95</td>
</tr>
<tr>
<td>5.2.1 Classification Problem Types</td>
<td>95</td>
</tr>
<tr>
<td>5.2.2 Attributes of a Hierarchical Classification Model</td>
<td>96</td>
</tr>
<tr>
<td>5.2.3 Types of Hierarchical Classification Frameworks</td>
<td>98</td>
</tr>
<tr>
<td>5.2.4 A closer look at Local Classifier Per Node frameworks</td>
<td>101</td>
</tr>
<tr>
<td>5.2.5 Hierarchical Performance Metrics</td>
<td>106</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>The Bayesian Network Interpretation</td>
<td>107</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Related Work</td>
<td>108</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Bayesian Networks</td>
<td>110</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Local Classifier Per Node as a Bayesian Network</td>
<td>111</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Practical Implications</td>
<td>115</td>
</tr>
<tr>
<td>5.4</td>
<td>Empirical Application: Methodology</td>
<td>124</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Hierarchical Classification with BENTHOZ-2015</td>
<td>124</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Training Local Classifiers</td>
<td>127</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Prediction Frameworks</td>
<td>130</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Local Classifier Performance</td>
<td>132</td>
</tr>
<tr>
<td>5.4.5</td>
<td>Hierarchical Performance</td>
<td>132</td>
</tr>
<tr>
<td>5.4.6</td>
<td>Abundance Estimates</td>
<td>133</td>
</tr>
<tr>
<td>5.5</td>
<td>Empirical Application: Results and Discussion</td>
<td>134</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Local Classifiers</td>
<td>134</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Hierarchical Performance</td>
<td>139</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Abundance Estimates</td>
<td>141</td>
</tr>
<tr>
<td>5.6</td>
<td>Conclusion</td>
<td>145</td>
</tr>
<tr>
<td>6</td>
<td>Hidden Node Bayesian Networks for Hierarchical Classification</td>
<td>149</td>
</tr>
<tr>
<td>6.1</td>
<td>Background: Hierarchical Bayesian Aggregation</td>
<td>149</td>
</tr>
<tr>
<td>6.2</td>
<td>Hidden Node Bayesian Networks</td>
<td>151</td>
</tr>
<tr>
<td>6.3</td>
<td>Methodology</td>
<td>152</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Hidden Node Bayesian Network Structure</td>
<td>152</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Training Local Classifiers</td>
<td>153</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Training the Network</td>
<td>156</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Computing Inference</td>
<td>158</td>
</tr>
<tr>
<td>6.3.5</td>
<td>Network Configurations</td>
<td>158</td>
</tr>
<tr>
<td>6.3.6</td>
<td>Comparing Performance</td>
<td>159</td>
</tr>
<tr>
<td>6.3.7</td>
<td>Regional Tuning</td>
<td>160</td>
</tr>
<tr>
<td>6.3.8</td>
<td>Sensor Redundancy</td>
<td>161</td>
</tr>
<tr>
<td>6.4</td>
<td>Results and Discussion</td>
<td>161</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Training Local Classifiers</td>
<td>161</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Network Structure</td>
<td>164</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Training the Network</td>
<td>164</td>
</tr>
<tr>
<td>6.4.4</td>
<td>Comparing Performance</td>
<td>166</td>
</tr>
<tr>
<td>6.4.5</td>
<td>Regional Tuning</td>
<td>170</td>
</tr>
<tr>
<td>6.4.6</td>
<td>Sensor Redundancy</td>
<td>172</td>
</tr>
<tr>
<td>6.5</td>
<td>Conclusion</td>
<td>173</td>
</tr>
</tbody>
</table>

## 7 Conclusion | 175  |
| 7.1     | Summary of Contributions | 175  |
| 7.1.1   | Hierarchical classification of benthic imagery | 175  |
| 7.1.2   | BENTHOZ-2015 Public Data Set | 176  |
| 7.1.3   | Bayesian Networks for Hierarchical Classification | 176  |
| 7.1.4 | Hidden Node Bayesian Networks | 177 |
| 7.2  | Extensions, Applications and Future Work | 178 |
| 7.2.1 | Multimodal sensor inputs | 178 |
| 7.2.2 | Enhanced local classifier performance | 179 |
| 7.2.3 | Adaptive Planning | 180 |
| 7.2.4 | Semantic and Visual Similarity | 183 |
| 7.2.5 | Biodiversity Metrics | 183 |
| 7.2.6 | Changing hierarchy over time | 184 |

**Bibliography**

187

**Nomenclature**

197
Abstract

Hierarchical Classification of Scientific Taxonomies with Autonomous Underwater Vehicles

Michael Stuart Bewley
The University of Sydney

Autonomous Underwater Vehicles (AUVs) have catalysed a significant shift in the way marine habitats are studied. It is now possible to deploy an AUV from a ship, and in a matter of hours capture tens of thousands of precisely georeferenced stereo images of the sea floor, along with other sensor data. A growing body of research is investigating ways to automatically apply semantic labels to this data. The two challenges motivating this research are as follows. First, a marine scientist requires semantic information in order to understand habitats and distributions of species, not terabytes of raw data. They currently spend extended periods of time manually labelling subset of images as a necessary precursor to performing scientific analysis. Second, the state of the art with AUVs typically involves pre-defining simple missions based on geographic coordinates. Defining suitable locations and missions is highly error prone, as the AUV covers a relatively small area of the ocean floor, and a frequent situation is the mapping of a reef or kelp bed surrounded by a large amount of sand. Combined with the high operating cost of an ocean research vessel, and the inability to communicate high bandwidth data such as images to an untethered vehicle, it would be desirable for the AUV to make real-time decision based on the same semantic content defined by marine scientists.

Much of the existing literature on the automated interpretation of benthic images has focussed on data sets from contiguous geographic areas, and performing supervised machine learning to predict a small number of mutually exclusive classes. The situation that these approaches typically fail to address is reuse. For the post hoc labelling by the marine scientists, it would be desirable to have a single automated annotation system that could adapt to any scientific interest, whether that is studying the distribution of *Ecklonia radiata* (a particular species of macroalgae), or identifying different morphologies of coral. Further, it would be desirable to make the system able to learn from the specific annotations already made by a wide variety of groups with different research interests, rather than requiring a specially constructed annotation set for each new problem. Finally, an AUV should be able
to be deployed in the field to make live semantic decisions (commonly referred to as adaptive planning), which implies minimal effort required in manual annotation, algorithm training and adjustment for each new deployment. One solution to this problem lies within the separate field of hierarchical classification; in which machine learning algorithms are trained to predict not just a binary or multi-class outcome, but a hierarchy of related output labels which are not mutually exclusive, such as a scientific taxonomy.

First, we investigated the implications of constructing a single species classifier (for *Ecklonia radiata*) on what was at the time a relatively large data set including a dozen missions from a geographic region in Tasmania. After investigating appropriate feature extraction and classification techniques for benthic images, we reinterpreted the problem as hierarchical classification, in order to predict labels within a pre-specified scientific taxonomy of 19 classes. These first two chapters (published in peer-reviewed conference proceedings) lay the foundation for the remainder of the thesis.

A richer data set was then constructed by creating a collaboration between five Australian research groups, and re-mapping existing labelled data to an agreed scientific taxonomy. The data set consists of more than 400,000 point labels on 10,000 images, conforming to a hierarchy of around 150 classes. Geographic regions include Tasmania, New South Wales, Queensland and Western Australia. This chapter, along with the data itself, has been published as an open data set in the *Scientific Data* journal of the Nature Publishing Group. It is hoped that the open data set will ease the path for future research in both the benthic imagery and hierarchical classification fields, by providing a substantial pre-assembled data set that is ready to use.

A review of the hierarchical classification literature revealed that a frequent approach to the problem involves training a set of individual classifiers for nodes in the class hierarchy, and assembling them via a number of heuristic algorithms to predict according to a hierarchy of labels. We identified an opportunity to apply Bayesian Network theory to improve the theoretical basis on which such a hierarchy of classifiers could be combined, which leads to a number of practical and novel improvements (such as the re-calibration of classifier probabilities to accurately represent the probability of class membership, given an example is a member of the parent class).

In the final chapter, we explore more complex Bayesian Networks which have further benefits, particularly for a robotics application, such as the ability to easily include and combine sometimes-present sensor data, and tune the system for performance in a particular geographic region without retraining the individual classifiers.
Acknowledgments

Going back to university after working was a substantial change, and had to fulfil three personal goals:

1. Allow me to practice machine learning on a large amount of real data, in an environment with talented experts to learn from;
2. working on a serious, practical problem with real world consequences;
3. be as enjoyable as it was challenging.

I knew that working with the marine robotics group at the Australian Centre for Field Robotics (ACFR) would fulfil all of these, but I never guessed to what extent. Firstly, I want to thank all those at the ACFR who made it possible for me to come along and focus almost entirely on machine learning, without the hard work of building and running a reliable robotics platform. Thanks to the technical staff (particularly Christian and Andy), who not only keep the vehicles running, but had the patience to support me on field trips, and getting my code running on the vehicles. Thanks to all those who designed the vehicles in the first place; students, post-docs, and above all to Stef and Oscar for making it all happen. As my supervisors, you did everything from give me guidance on my topic, to careful reviews of my papers, to giving me the opportunities to have truly once-in-a-lifetime experiences (like eating fresh caught sushi at the end of a long campaign in Tasmania, snorkelling alongside an AUV off a remote tropical island, and joining E/V Nautilus deep sea exploration expeditions to the Mediterranean and Caribbean). Thanks a million times for taking me on as a student, and for getting me through to the end. Likewise to the folk at the Ocean Exploration Trust, working on board the Nautilus was unforgettable.

Bart, thanks for taking me under your wing to guide me through my first conference paper, and defining my initial topic. Navid, thanks for working with me on my second, and for stepping in to present the paper for me when I was ill. I’d also like to thank Mitch and Fabio, for supporting me through my annual reviews. No thanks would be complete without reference to all the other students (you know who you are), for the extensive discussions, support, advice, and camaraderie.

Ari, Renata, Nicole, and all the other authors on BENTHOZ-2015, thanks for your support and hard work in getting such a large, rich data set into the public sphere. I certainly didn’t expect to be exchanging emails with the editor of a Nature journal as a student, but I sincerely believe we built something that future students and researchers can build on to rapidly make advances that we can only dream about.
I also owe thanks to two great companies for indulging me by letting me work part time through my studies. Thanks to Robert, Rami and Adam for letting me continue my work at Cochlear as I started my PhD, and to Rob and Rachel for letting me join them at the Commonwealth Bank part time while I finished it.

Dad, thanks for telling me not to do a PhD unless I had a good reason, and a project I was passionate about. I found one, and don’t regret it for a second.

Lastly, to Fliss (my gorgeous wife); thanks for supporting me through the good times and the tough, and for working so hard as you brought our children into the world. I couldn’t have gotten through those years without such a loving family to come home to each day.
1 Introduction

1.1 Overview

The aim of this thesis is to lay the foundations for a reusable, automated semantic classifier of autonomous underwater vehicle (AUV) data. The motivation for creating a system to automatically annotate the data captured by an AUV during an environmental survey is twofold. Firstly, the manual effort required for a marine scientist to label content in the retrieved images is untenable. With tens of thousands of images returned from a deployment lasting only a few hours, a marine scientist might spend 5 minutes identifying the content at dozens of points in each image; as a result, it is common for only 1% of the images from a campaign (series of AUV deployments) to be labelled after several years of work. The second motivation is in the potential application of such a system to real-time classification by the AUV during a deployment. At present, state of the art solutions involve an entirely pre-planned route for the vehicle, defined in terms of geographic coordinates on the sea floor. On board classification of sensor data would open the way for adaptive missions that adjust the path of the AUV in response to the same semantic interpretation as the marine scientists; doing so could greatly increase efficiency of operations in the field, from simple checking of assumptions that a planned mission is operating on a desirable benthic habitat, to self-optimising missions that map out semantically
defined areas of interest (such as a patch of reef), or coordination of multiple AUV deployments based on real-time feedback of a new area being explored.

Prior work on automated classification of benthic images typically involves detection of a small number of classes (usually only one), representing a carefully constructed solution to a pre-defined scientific problem (such as identifying crown of thorns starfish [26], or a particular species of kelp [33]). Further, testing usually only consists of “in-sample” validation; when generalisation performance of a predictive model is tested either by cross-validation on a data set, or reserving a random subset of the training data for testing. “Out-of-sample” validation is when a performance is tested on a completely separate data set; in time-series modelling, this is usually data taken from a later point in time. For benthic data, this should also include new geographic locations. Both of these aspects of current research result in models that cannot be applied beyond the immediate context (problem and data set) in which they were constructed.

1.2 Problem Statement

This thesis is concerned with the identification of, and solution to, challenges in creating a reusable automated classifier that can meet a wide range of scientific goals, and be applied to data from new geographic locations with minimal readjustment. The motivation is to reduce the burdensome manual labelling effort of benthic imagery by marine scientists, and permit an AUV to robustly perform live semantic interpretation of its surroundings when deployed in the field.
1.3 Contributions

The core contributions of this thesis are as follows:

1. **Framing automated interpretation of benthic imagery as a hierarchical classification problem:** Re-framing the detection of species in benthic AUV data as a hierarchical classification problem, to permit the combination of supervised training data from a number of sources, and flexibility in applying the classifier to a wide range of scientific problems.

2. **BENTHOZ-2015 public data set:** Collating and publishing a new public data set based on over 400,000 expert annotated points on approximately 10,000 images, conforming to a label hierarchy of 148 classes, with good rep-
presentation from a 5 year span around the Australian coastline. This data set will be pivotal in permitting the machine learning, computer vision and hierarchical classification communities to contribute to solving the problem, and compare results on a real world data set. It also has benefits for the marine science community more generally, as the raw data from studies is not typically made publicly available.

3. **Bayesian Networks for hierarchical classification**: Contributing to the hierarchical classification literature by introducing Bayesian Network theory to explain assumptions and drive choices in a space dominated by heuristic solutions and empirical evaluation.

4. **Hidden node Bayesian Networks**: Defining a more advanced Bayesian Network framework (referred to here as a *hidden node Bayesian Network*) for hierarchical classification, which advances the literature with a new, theoretically grounded approach that applies well to classifying hierarchies in environmental monitoring robotics. Its key advantages are that it can be easily adjusted by a non-expert for anticipated class distributions and classifier performance (e.g. in a new geographic area), and include evidence from multiple distinct classifiers which may or may not be present (e.g. one based on image texture, and another based on an AUV’s depth sensor). The framework is intended to easily support many future performance improvements that use more advanced image features and classifiers (e.g. a “deep learning” approach to generating image features), the inclusion of classifiers based on different sensor data (such as acoustic backscatter, hyperspectral imaging or dense stereo for fine scale 3D structure), and even expert defined classifiers to distinguish rare species where training data is scarce.
Chapter 1 introduces the thesis, outlines its core contributions, and provides the background context that underpins the subsequent chapters.

In Chapter 2, we begin with the motivating example of automatically recognising a single species (*Ecklonia radiata*, or kelp) based on benthic images captured by an AUV. We explore various practical aspects of the problem, including the utilisation of expert labelled images in a supervised learning framework, evaluation of a range of commonly used methods for computing descriptors for labelled points, and exploration of the variability induced by training and testing across different deployment sites within a larger geographical area (South Eastern Tasmania). As well as providing useful background and review of the literature, the investigation of species identification on such a large data set is novel, particularly in the exploration of how well the classifier is likely to perform if applied to a new dive site. This work was published as [12], winning the prize for best student paper at the Australasian Conference on Robotics and Automation in New Zealand in 2012.

Chapter 3 moves on to reframe the problem for classifying not just the presence or absence of an individual species, but an entire scientific taxonomy of 19 different classes using the same data set. The work extends from [12], seeking to create a framework that permits a single, unified classification system to be trained and then reused for a range of problems, in a variety of geographical locations. This is markedly different from the bulk of works in the benthic automated detection literature, which focus on a single species, and evaluate performance on a random sample drawn from the same data set on which the algorithms have been trained. This chapter crystallises the fundamental goal of the thesis: To identify and overcome the challenges in creating a reusable automated classifier that can meet a wide range
of scientific goals, and be applied to data from new geographic locations with minimal readjustment. The motivation is to reduce the burdensome manual labelling effort of benthic imagery by marine scientists, and permit an AUV to robustly perform live semantic interpretation of its surroundings when deployed in the field. This work was published as [14], and presented at both the Fine Grained Visual Classification workshop of the Computer Vision and Pattern Recognition (CVPR) conference in Portland, USA, and the the Field and Service Robotics (FSR) conference in Australia in 2013.

Chapter 4 takes a slight tangent to prepare “BENTHOZ-2015”, a more comprehensive data set for subsequent investigations, and future research. Rather than looking at data from a single year in Tasmania (with approximately 60,000 labelled image points and a hierarchy of 19 classes), data from four major research groups in Australia was combined, spanning 5 years, large portions of the Australian coastline, more than 400,000 labelled image points, and a hierarchy of 148 classes. This level of collaboration, and willingness to publish an open access data set was unprecedented in the field. The core contribution of the work is to permit machine learning and computer vision researchers to explore hierarchical classification on a large, real world data set, and compare results with other approaches on a common benchmark. This work was published as [13] in Nature’s Scientific Data journal in 2015.

Chapter 5 conducts a review of the hierarchical classification literature, and identifies that the majority of frameworks are heuristic in nature, and assessed almost exclusively on empirical performance. The core contribution of the chapter is to introduce a theoretically grounded framework for hierarchical classification based on Bayesian Networks, which supports a particular approach to one of the more common heuristic algorithms, and clarifies the theoretical implications of a number of
the assumptions and choices which are made. The work demonstrates the proposed approach using the data set constructed in Chapter 4.

Chapter 6 takes the application of Bayesian Networks to hierarchical classification beyond the heuristic approaches, to a novel approach that excels at several properties required to meet the goals of the thesis. Individual classifiers can be trained for classes in the hierarchy, and minor adjustments made to account for the anticipated distribution of classes in a new geographic area. Other sensor data (such as depth information for an AUV), can be used to improve classification performance, or ignored when the sensor is unavailable. The operation of the classifier is also easily interpreted, which is important to gain the trust of marine scientists, and support effective troubleshooting during field deployment.

Chapter 7 summarises the contributions, and identifies opportunities for future research directions that have been made possible by this work.
2 Single species classification on a large data set

This chapter was presented as a conference paper entitled *Automated species detection: An experimental approach to kelp detection from sea-floor AUV images* at the Australasian Conference on Robotics and Automation in 2012, receiving the “Best Student Paper” award [12]. It is reproduced verbatim here, as it introduces a number of the key concepts on which the remainder of this thesis are developed:

- A review of the prior work done in the field of automated annotation of benthic images

- Establishing the need for larger, more diverse data sets in order to work towards a practical system to recognise content in AUV images.

- The high degree of variability between AUV deployments even within the same campaign, and the resulting challenges for an automated classification system.

- An approach to using expert labelled points within images as the basis for performing supervised learning to detect a single species, rather than the previous trend of whole-image classification of more general habitats.

*Contribution: I was the primary author of the conference paper, and received support, guidance and other input from the other authors listed in the citation* [12].
Abstract: This paper presents an experimental study of automated species detection systems suitable for use with Autonomous Underwater Vehicle (AUV) data. The automated detection systems presented in this paper use supervised learning; a support vector machine and local image features are used to predict the presence or absence of *Ecklonia Radiata* (kelp) in sea floor images. A comparison study was done using a variety of descriptors (such as local binary patterns and principal component analysis) and image scales. The performance was tested on a large data set of images from 14 AUV missions, with more than 60,000 expert labelled points. The best performing model was then analysed in greater detail, to estimate performance on generalising to unseen AUV missions, and characterise errors that may impact the utility of the species detection system for marine scientists.

2.1 Introduction

Autonomous Underwater Vehicles (AUVs) are being increasingly used to support environmental monitoring programs [37], particularly in the area of monitoring changes to sea-floor (benthic) ecosystems. Benthic ecosystems have been identified as having key importance in understanding how the marine environment responds to various pressures, including fishing activities, invasive species and climate change [6, 97].

The increased use of AUVs is primarily due to their ability to routinely capture high quality geo-refenced imagery of the sea floor in an efficient and repeatable manner. An AUV such as *Sirius* [97] is capable of performing missions autonomously. Over several hours, tens of thousands of images of the sea floor can be captured (along with other sensor data). As part of Australia’s Integrated Marine Observation System (IMOS), the AUV IMOS Facility was established to collect repeated measurements of oceanographic data from reference sites around Australia’s coastal areas [97]. The
2.1 Introduction

data set used in this paper is from a single campaign at a reference site in Eastern Tasmania, conducted in October 2008 [5, 96].

AUVs are enabling researchers to gather visual data with a precision and volume that was previously impossible. The process of manually identifying and labelling species within AUV images is highly labour intensive; in the data set used in this paper, 50 random points were labelled in each image. Understandably, this process is only feasible on a small fraction of the available data set. Scientists use these labels to estimate coverage of species or substrates within an area. There are two natural goals for an automated species detection system: firstly, to reduce or eliminate the manual workload; secondly, to make use of the complete data set in analysis, rather than being restricted to these small subsets.

In this paper, we present results on a series of experiments on automated species detection systems for *Ecklonia Radiata* (kelp). The experiments have two main phases:

- A comparison study was performed on the use of local image descriptors to predict the presence of kelp. Several local image descriptors were used at a range of scales to train a support vector machine. Results from these models act as a performance benchmark for kelp detection on a large AUV data set.

- The highest performing model was analysed in further detail to estimate the performance when used on unseen AUV dives, and determine the distribution of prediction errors across some environmental variables. Given that the goal is to build species detection systems that apply to the analysis of the ecology and biology of an area, it is important to estimate the likely performance in a variety of scenarios, rather than simply maximising the classification accuracy on the whole test set.

While the long term aim is to detect multiple species using the AUV data, detection
Chapter 2 Single species classification on a large data set

of kelp is useful in its own right. Around much of Australia’s sub-tropical and temperate areas, kelp forests act as the primary habitats for many species, and the extent of kelp is used as a proxy for the health and biodiversity of an area [89].

The remainder of this paper is divided into the following sections: Section 2.2 describes related work done in the area of classification and clustering on benthic images. Section 2.3 describes the algorithms used to detect kelp in this paper. Section 2.4 is an overview of the way the AUV images were collected and labelled, as well as the training and testing strategy. Section 2.5 compares results from a range of descriptors and scales. Section 2.6 takes the best performing model from the previous section, and subjects it to further examination. Section 2.7 provides conclusions and directions for future work.

2.2 Related Work

While this paper frames species detection in the context of the scientific end users of the data, there is also a clear motivation from a direct robotics perspective. Given the limited and unreliable communication bandwidth with AUVs, it is desirable for an AUV to make sense of its surroundings, and respond appropriately. Existing AUV missions typically rely on following pre-determined paths, utilising an underwater simultaneous localisation and mapping (SLAM) solution [95]. A long term research goal is for the robot to associate semantic labels with its surroundings as it travels, such that it can adjust its mission, or send the information to human operators.

2.2.0.1 Whole-Image Approaches

An initial approach that goes some way towards species detection lies in performing clustering and classification at the whole image level. The scale of the images
produced by an AUV is fairly predictable (around 1-2 metres across). This is because strong attenuation of light in water requires the robot to travel at an altitude of around 2 metres above the sea floor. Work following the whole-image approach includes online classification of image habitats for adaptive sampling [74], active learning to improve semantic relevance of image clusters [44], and post-processing of large quantities of AUV images for unsupervised clustering [87, 71].

2.2.0.2 Sub-Image Approaches

The whole-image approaches provide useful semantic information on the image data, but they lack the ability to discriminate between smaller features. As shown in Figure 2.1, an image can contain examples of multiple occurrences of multiple species. For species detection, it is more accurate to label each occurrence of each species.

In a recent paper, a solution to sub-image detection of coral species from images is presented, with a similar aim of providing coverage estimates for marine scientists [9]. The authors highlight the challenge of recognising amorphous and variable biological organisms, compared to the structured environments often used to test object detection algorithms. In addition, the type of semantic labelling (random image point labels) is the same as presented here.

The work performed in [33] is particularly relevant, as it aims to detect exactly the same species of kelp as our study, and is also based on AUV data. In this work, image features were combined with clustering to produce a predictive model, and results were assessed against a set of 45 images. Our paper expands further, using a wider range of image features on a significantly larger data set.

Other sub-image approaches have also been attempted. Several groups have created detection mechanisms for particular species, such as starfish [84, 26, 34] and coral
Figure 2.1: An example AUV image containing multiple dominant species. Expert labels are overlaid (with KELP as ECK).

[85]. These studies have typically only tested a small range of features. In addition, the available data was often insufficient to either produce optimal results, or evaluate classifier performance sufficiently to enable confident use of its output in a marine science context.

Some key distinctions exist between the aim of these earlier studies, and the current paper. The data set introduced in this paper presents an opportunity not currently present in the literature. With more than 60,000 labelled points in the set, only the coral data set of [9] is comparable in size (it is actually larger, at 400,000 labelled points). The data set from that study was, however, captured by divers using hand-held cameras, rather than an AUV. The potential advantage of using an AUV data set is significant: the ease of capturing a comprehensive data set; spatial registration of the images; availability of additional sensor data (such as 3D stereo) which may
boost performance; and the ability to perform much more precise repeated surveys. In addition, the data set used here is from temperate coastal waters off Tasmania, rather than tropical reef images from French Polynesia.

2.3 Methodology

This methodology applies to both the comparison study (Section 2.5) and the final model evaluation (Section 2.6). In summary, experts hand-labelled randomly distributed keypoints on AUV images. Image descriptors were computed on local image patches at various scales centred around the keypoints. Note that throughout this paper, we refer to a feature as a particular type of descriptor computed at a particular scale. Each set of features was used as the input to supervised classification. The goal was to learn to classify the presence or absence of kelp, based on the expert label assigned to the centre pixel of each patch.

2.3.1 Scale and Keypoints

All the features used for training and testing in this study were derived from local image patches. Square patches of various sizes were used, with a patch centred on each hand labelled keypoint from the data set. This was done to provide some local context around the centre pixel, from which features could be derived. A range of scales were evaluated in the experiments, from $7 \times 7$ to $95 \times 95$ pixel patches. The image resolution is $1360 \times 1024$. At 2m above the sea floor, these patch sizes correspond to approximately 0.7cm to 10cm. Scale trades off between precision (as only the centre pixel is being labelled), and the inclusion of wider context.
2.3.2 Descriptors

Based on the square patches defined above, various descriptors were used to describe the keypoints. This dimensionality reduction process (where the NxN image patch is reduced to a lower dimensional feature vector) is a commonly used step in machine learning problems; it projects the data into a new, lower dimensional space before the classification algorithm is applied.

2.3.2.1 Raw Pixels

For the smallest patch sizes (limited by computational tractability), the patch pixels were used directly as a feature vector. As all descriptors are derived from these raw pixel values, it is worthwhile comparing the results.

2.3.2.2 Principal Component Analysis (PCA)

PCA selects an orthogonal set of coordinates that maximise variance [16]. In the experiments, PCA was applied to a flattened vector of the raw pixel values from each patch, and the lowest variance components were discarded. By experimentation, it was found that using anywhere between 20 and 60 components had little impact on the results. As such, 30 components were selected, and whitening was used in all cases. Greyscale PCA refers to PCA performed on the means of RGB channels. RGB PCA refers to PCA performed directly on all three colour channels.

2.3.2.3 Grey Level Co-occurrence Matrices (GLCM)

GLCMs were first proposed by Haralick in the 1970s [49, 48]. Where PCA is a naive, commonly used dimensionality reduction technique, GLCMs were created specifically for forming descriptors that represent texture in greyscale images, and
2.3 Methodology

have also been extended to colour. The colour GLCM extension was used in [33] for kelp detection. In that study, the GLCM parameters of uniformity, contrast, correlation, local homogeneity and entropy were used. These parameters were computed at angles of 0, 45, 90, 135 degrees and summed to obtain one final rotation invariant matrix. These statistics were computed at four radii, spaced approximately logarithmically between a single pixel offset, and half the patch width (50 × 50 or 100 × 100 pixels). To reduce the dimensionality of the matrices, images were reduced to 32 intensity levels on each channel. In order to make use of colour information, the GLCMs were computed between pairs of colour channels (although the red channel was not used, due to the strong absorption of red light in water). By using the same GLCM descriptor in this paper (however at different scales), we evaluated it on a much larger data set, against other common descriptors.

2.3.2.4 Local Binary Patterns (LBP)

Local Binary Patterns (LBP) [70] are a texture descriptor that has been widely used for underwater image interpretation. [64, 85, 26, 44]. They can be computed at multiple scales and made rotation invariant. Combining multiple LBP scales has proven useful for improving classification performance [70].

A number of studies have been performed comparing LBPs to other descriptors (including GLCMs), and LBPs were usually found to provide superior results [42, 79, 26, 86].

In brief, a ring of pixel intensities are compared to the intensity at the centre of the ring. A threshold is applied to the differences, to create the local binary pattern. A histogram of the occurrence of these patterns is computed for each image patch, and used as the descriptor. In this study, a ring radius of 1 pixel, with 8 points, was used to create the patterns.
2.3.2.5 Histogram of Oriented Gradients (HOG)

The HOG algorithm was designed for detection of humans in images, but has also been used in other applications [29]. It breaks an image patch into cells, and computes gradients within those cells. It also uses blocks (adjacent cells) to perform some relative normalisation for illumination invariance. For this application, HOG with $7 \times 7$ cells was used, with a block consisting of $3 \times 3$ cells. This means that the smallest patch used ($7 \times 7$) consisted of a single cell, whereas the largest patch ($95 \times 95$) consisted of a grid of $13 \times 13$ cells. As the descriptor was produced by concatenating the cell histograms, the length of the descriptor increased with the square of the patch edge size.

2.3.3 Classification

The key focus of this paper is on a comparison of features, and the system level solution of the kelp detection problem. A single classifier was used - a Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel. While further optimisation may be possible using alternative classification algorithms, RBF-SVMs are known to provide good results on a wide range of problems [51]. In brief, an SVM finds the globally unique linear separating hyperplane that separates the two classes with the maximal margin. As an extension, the RBF kernel first transforms the input vector to infinite dimensional Hilbert space, so that non-linear separations can be obtained. Only two parameters need to be tuned. $C$ is the cost parameter (large $C$ is less tolerant of non-separable data sets, and forces a model that is more accurate on the training set). $\gamma$ adjusts the radius of the kernel (large $\gamma$ decreases the kernel radius, which reduces the distance over which support vectors have influence, also creating a more complex decision boundary).
2.4 Data Set

In this study, parameter selection was performed by observing cross-validation results of a grid search over $C$ and $\gamma$. Following the guidance of [51], the parameter ranges used were $C = [2^{-5}, 2^{15}]$, and $\gamma = [2^{-15}, 2^3]$, with a logarithmic density of powers of 2.

2.4 Data Set

An additional objective of this paper is to establish a benchmark for species detection in a large, high quality AUV data set. Future research can then assess the benefit of using AUV sensors and positioning, compared to image processing alone.

The primary use of the labelled data set used in this study is to estimate percentage cover and distribution of various species and features in an area. For kelp, this is done by taking all the labelled images from the area of interest, and calculating the percentage of labelled points that are labelled as $ECK$.

2.4.1 Classes and Labelling

The entire data set is comprised of 14 dive missions conducted by the AUV $Sirius$ off the South-East coast of Tasmania in October 2008 [96]. To capture images, Sirius uses a calibrated stereo pair of 1.4 megapixel cameras, and uses strobes to illuminate each capture. From the data set containing over 100,000 stereo pairs of images, marine scientists selected every 100th colour image, and used the CPCe software package [58] to label 50 random points on each (as shown in Figure 2.1).

A wide range of class labels were used, indicating biological species (including types of sponge, coral, algae and others), abiotic elements (types of sand, gravel, rock, shells etc.), and types of unknown data (ambiguous species, poor image quality, etc.).
Precise details of the labelling methodology can be found in [5]. The KELP (ecklonia radiata) class was used as-is, and all other points were relabelled as OTHER. These labels were then used as training and testing instances, with features derived from localised image patches. Many of the classes can be arranged in a hierarchy, as shown in Figure 2.2. Of particular relevance are the groups “Brown Macroalgae” (containing kelp and other brown aquatic plants), and “Macroalgae” (containing brown, red and other macroalgae). Figure 2.3 shows some examples, where the yellow crosses indicate the labelled points, with the surrounding patch on which the descriptors were computed.

![Class hierarchy summary](image)

**Figure 2.2:** Class hierarchy summary

### 2.4.2 Training, Validation and Test Sets

As shown in Figure 2.4, data was divided into training, validation and test sets. This is described in more detail below.
2.4 Data Set

2.4.2.1 Test Set

A random 20 percent of the images were reserved for use in evaluating the final model. Data from these images was not used in any stage of the model selection or parameter optimisation.

2.4.2.2 Training and Validation Sets

The images not reserved in the test set were used for training and validation. 67% of the randomly labelled image points were used as the Training Set, and the remaining 33% as the Validation Set. A fixed validation set was used for model comparison, instead of cross validation. This greatly reduced the computational requirement (by approximately an order of magnitude compared to 10-fold cross validation). Training set sensitivity results on the final model (Figure 2.7) demonstrated that peak performance could be obtained with significantly smaller training sets, rendering cross-validation unnecessary.

Figure 2.3: Example 95×95 patches of kelp (left) and all classes (right)
2.4.2.3 Training and Testing Strategy

For the Model Comparison stage, 3-fold cross validation was performed within the Training Set to optimise the SVM parameters $C$ and $\gamma$. Models were then retrained on the full training set, and performance results reported on the Validation Set. Once the best model had been selected, detailed analysis was performed on the Test Set results.

2.4.3 Dive Characteristics

Figure 2.5 shows the most prominently featured classes in each dive. It is worthwhile noting: Overall, only around 5% of the points are KELP; the largest component of OTHER is sand; 6 of the dives included no KELP at all. While dives in the campaign
2.5 Model Comparison Experiments

were conducted in a similar geographic area (the South-East coast of Tasmania), and at a similar time (over several weeks) they vary greatly in depth and content. It is important to train and test across multiple dives, as the detection system needs to be robust enough to manage anything from a deep dive over sand, to a shallow dive with rock and kelp.

2.5 Model Comparison Experiments

A series of experiments was run to test the performance of classifiers trained using the various local image descriptors and scales described in Section 2.3.

2.5.1 Patch, Feature and Classifier Combinations

2.5.1.1 Greyscale Comparison

A classifier was trained to predict KELP vs OTHER, using the descriptors extracted from greyscale patches at each scale. This allows both a comparison of the descriptors, and an assessment of which scales are most useful.

2.5.1.2 Colour Comparison

Some descriptors (Raw, PCA, GLCM) can be extended to work in the RGB colour space. This comparison is less complete (as it includes fewer descriptors), but it allows an assessment of how useful the colour information in the AUV images is used for species detection. Note that for colour GLCM, only the green and blue channels were included, to maintain consistency with [33].
2.5.2 Performance Evaluation

One key attribute of the data set is that it is significantly skewed against KELP. The OTHER class contains approximately 20 times the number of instances. During classification, the instances were weighted \[52\] in the RBF-SVM, so as not to bias the solution against the minority class.

Class weighting reduces the inherent classifier bias, but it is still important to choose an appropriate performance metric for the classifiers, that is insensitive to class imbalance. The f1-score is commonly used for biased classes, and is used as the primary performance metric in this paper. It is defined as the geometric mean of precision and recall on the minority class. Precision (the percentage of instances classified as kelp that are actually kelp) and recall (the percentage of genuine kelp that is classified as kelp) must both be high in order to obtain a good f1-score.

2.5.3 Results

Table 2.1 and Table 2.2 are the final results on the Validation Set, after training the RBF-SVM classifier on each descriptor and scale. Note that where a '-' is listed, the model was unable to be run, usually due to dimensionality of the feature vector, or memory limitations. Figure 2.6 shows both greyscale and colour descriptors across all scales.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Raw</th>
<th>PCA</th>
<th>GLCM</th>
<th>HOG</th>
<th>LBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 x 7</td>
<td>0.63</td>
<td>0.63</td>
<td>0.41</td>
<td>0.17</td>
<td>0.38</td>
</tr>
<tr>
<td>15 x 15</td>
<td>0.60</td>
<td>0.61</td>
<td>0.48</td>
<td>0.30</td>
<td>0.54</td>
</tr>
<tr>
<td>31 x 31</td>
<td>-</td>
<td>0.57</td>
<td>0.51</td>
<td>0.44</td>
<td>0.61</td>
</tr>
<tr>
<td>63 x 63</td>
<td>-</td>
<td>0.54</td>
<td>0.57</td>
<td>-</td>
<td>0.62</td>
</tr>
<tr>
<td>95 x 95</td>
<td>-</td>
<td>0.57</td>
<td>0.63</td>
<td>-</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Table 2.1: f1-scores for greyscale descriptors
2.5 Model Comparison Experiments

<table>
<thead>
<tr>
<th></th>
<th>Raw</th>
<th>PCA</th>
<th>GLCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 × 7</td>
<td>0.71</td>
<td>0.69</td>
<td>0.48</td>
</tr>
<tr>
<td>15 × 15</td>
<td>-</td>
<td>0.73</td>
<td>0.60</td>
</tr>
<tr>
<td>31 × 31</td>
<td>-</td>
<td>0.68</td>
<td>0.65</td>
</tr>
<tr>
<td>63 × 63</td>
<td>-</td>
<td>0.66</td>
<td>0.67</td>
</tr>
<tr>
<td>95 × 95</td>
<td>-</td>
<td>-</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 2.2: f1-scores for colour descriptors

2.5.4 Discussion

2.5.4.1 Greycale Comparison

Both Raw pixels and PCA performed best at the smallest scale (7 × 7). The similarity indicates that at the smaller scales, very little useful information is discarded by PCA. As the scale was increased, training the SVM on raw pixels became computationally intractable, and PCA showed a slight decrease in performance. This decrease in performance at larger patch sizes for PCA suggests that the descriptor takes its information from the immediate local area, and larger patch sizes are simply adding noise, weakening performance.

HOG, GLCM and LBP all performed relatively poorly at smaller scales, with steadily increasing performance towards the larger patch size. As these descriptors were explicitly constructed to capture textural information, this strongly suggests that the textures that are best suited to kelp detection are on the scale of the larger patch sizes.

It is worth noting that the PCA model operating on very localised patterns performs similarly to LBP and GLCM at the largest patches (the LBP having a slightly higher performance). Given the difference in scale used, it is possible that a combination of the 7 × 7 PCA and 95 × 95 (or larger) LBP would have superior performance. This suggests that the combination of image descriptors at multiple scales could be an interesting area for further work. In addition, although LBP gave the best
greyscale performance, no colour extension was tested. A colour extension of LBP could provide superior results to the PCA and GLCM descriptors that were tested, and would be a worthwhile area of future work.

2.5.4.2 Colour Comparison

Colour correction in underwater imagery is a notoriously difficult problem. Water absorbs the red much more strongly than other parts of the spectrum, so there is strong dependence on depth of both illumination and colour. The strobes on the AUV are limited in range, and objects below the AUV can vary from well illuminated, to strongly over or under exposed.

Despite this, all of the colour descriptors exhibited a clear performance increase over their greyscale counterparts, at all scales. This indicates that even with the basic colour correction used in this data set \cite{54}, including colour information in the descriptors is beneficial for kelp detection. A method for more advanced colour correction has been proposed \cite{22}, which may result in further improvements to performance.

2.6 Final Model Evaluation

The primary aim of separating this study into “Model Comparison Experiments” and “Final Model Evaluation” was to enable an in-depth analysis of the quality and error on the final classifier. This section includes retraining the model on various subsets of the Training Set, and then an analysis of its properties on the Test Set.

The model chosen for evaluation was the PCA (RGB) at the scale of $15 \times 15$. This model was selected for its combination of high performance, and low computational load (only requiring the linear combination of a small image patch, where other
2.6 Final Model Evaluation

descriptors have greater computational complexity, and require computation over much larger patches).

2.6.1 Training Set Sensitivity Tests

In this group of tests, we retrained the classifier on subsets of the Training Set, to determine the impact on performance. Determining an appropriate training set size is of great interest for automating species detection. If the training data is found to be insufficient, more data could be manually labelled in order to improve performance. However, if a classifier of similar performance can be built with less training data, then this puts less burden on the scientists to manually label large amounts of data.

The size of the training set was varied in several ways:

1. Discarding a percentage of random patches. This tests whether the training set was of sufficient size to permit optimum performance on the model

2. A “single dive” approach. This trains on one dive at a time, and demonstrates how optimistic a single-dive model is when generalised to many dives

3. A “leave one out” approach. This trains on all but one dive, simulating the performance on an additional, unseen dive being performed in the same geographic area.

2.6.2 Error Analysis Tests

The experiments in Section 2.6.1 were designed to test whether the data sets were of sufficient size to provide good generalisation performance. The question remains, however, as to where the classifier makes mistakes. As stated previously, the primary
intended use of the kelp detection system is to estimate percentage cover of kelp in an area. Unevenly distributed error between dives, or relative to other variables, could result in coverage estimate errors that are significantly above what the overall model performance would suggest. Any variation in performance for some subset of the data (such as a bias for another type of macro-algae to be recognised as kelp) needs to be understood by scientists making use of the automated detection system, or their conclusions may be biased by the error of the detection system. To provide insight into this problem, the error on the Test Set was broken down by dive, and by groups of some of the original expert labelled classes.

### 2.6.3 Results and Discussion

#### 2.6.3.1 Overall Performance

<table>
<thead>
<tr>
<th></th>
<th>Training Set</th>
<th>Testing Set</th>
<th>Precision</th>
<th>Recall</th>
<th>f1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
<td>Training Set</td>
<td>0.79</td>
<td>1.0</td>
<td>0.88</td>
<td></td>
</tr>
<tr>
<td>Training Set</td>
<td>Validation Set</td>
<td>0.69</td>
<td>0.77</td>
<td>0.73</td>
<td></td>
</tr>
<tr>
<td>Training Set</td>
<td>Testing Set</td>
<td>0.64</td>
<td>0.74</td>
<td>0.69</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.3: Precision, Recall and f1-score on each testing set, using PCA (RGB) on 15 x 15 patches

Results are shown in Table 2.3. The f1-score of KELP decreases when assessed on the Training, Validation and Testing Sets. The higher performance on the Training Set could indicate slight over-fitting. The slightly lower performance on the Testing Set compared to the Validation Set could be attributed to one of two effects.

Firstly, the Testing Set contained keypoints from completely unseen images, whereas the Validation Set is made up of keypoints that occupy different areas of the same images as the training set. This makes the Validation Set more optimistic; unusual illumination, colour balance or content of an image is likely to be represented in
2.6 Final Model Evaluation

both Training and Testing Sets.

Secondly, there is a known effect when a large number of models are compared on a validation set. If the model with the best performance is selected, the performance quoted on that same set is biased to be slightly optimistic [68].

2.6.3.2 Training Set Sensitivity

The Training Set percentage results in Figure 2.7 demonstrate that the model has no significant performance improvement when trained on more than around 30% of the original Training Set. This supports the earlier assumption that the training set was of sufficient size that a separate Validation Set and Test Set could be reserved (rather than using cross-validation). It also gives an indication of how much hand-labelling is required to make an optimum model (at least one based on local image features). As discussed earlier, the onerous nature of hand labelling makes this a valuable contribution. 30% of the original Training Set corresponds to 10,086 hand labelled points (700 of which are kelp). This is far smaller than the full data set of more than 60,000 hand labelled points.

In Figure 2.8, the right hand columns show the performance one gets looking solely at data from one dive (both for training and testing). The left hand columns show the performance of the exact same model on the full Test Set (all dives). It is interesting to note a significant drop for most dives when tested on the full Test Set. The marked drop (of around 20% in kelp f1-score) show that generalisation from a single dive is poor even on dives done at a similar time, in a nearby geographic location.

The results after training on all but one dives (Figure 2.9) are very similar to the performance when the full training set is used.
The most interesting deduction comes from combining these results. Together, they suggest that although training on a single dive has poor generalisation, it is reasonable to expect good generalisation results in a geographic area with sufficient training data. This means that once a species detection system has been trained on a number of dive missions, it may not be necessary for experts to perform any further hand labelling on subsequent dives in the area.

### 2.6.3.3 Error Analysis

<table>
<thead>
<tr>
<th>Dive</th>
<th>Total Accuracy</th>
<th>f1-score</th>
<th>% Kelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>92.92%</td>
<td>80.15%</td>
<td>18.96%</td>
</tr>
<tr>
<td>12</td>
<td>91.63%</td>
<td>73.79%</td>
<td>17.52%</td>
</tr>
<tr>
<td>15</td>
<td>86.18%</td>
<td>60.66%</td>
<td>12.48%</td>
</tr>
<tr>
<td>20</td>
<td>94.74%</td>
<td>71.67%</td>
<td>9.12%</td>
</tr>
<tr>
<td>6</td>
<td>97.70%</td>
<td>87.10%</td>
<td>9.06%</td>
</tr>
<tr>
<td>5</td>
<td>93.53%</td>
<td>57.92%</td>
<td>5.55%</td>
</tr>
<tr>
<td>19</td>
<td>97.45%</td>
<td>69.84%</td>
<td>5.38%</td>
</tr>
<tr>
<td>7</td>
<td>98.95%</td>
<td>40.00%</td>
<td>0.53%</td>
</tr>
<tr>
<td>16</td>
<td>99.68%</td>
<td>-</td>
<td>0.26%</td>
</tr>
<tr>
<td>13</td>
<td>99.28%</td>
<td>-</td>
<td>0.00%</td>
</tr>
<tr>
<td>11</td>
<td>99.54%</td>
<td>-</td>
<td>0.00%</td>
</tr>
<tr>
<td>10</td>
<td>99.33%</td>
<td>-</td>
<td>0.00%</td>
</tr>
<tr>
<td>9</td>
<td>100.00%</td>
<td>-</td>
<td>0.00%</td>
</tr>
<tr>
<td>8</td>
<td>99.84%</td>
<td>-</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Table 2.4: Performance on each dive, using PCA (RGB) on $15 \times 15$ patches

<table>
<thead>
<tr>
<th>Test Set subset</th>
<th>N</th>
<th>Accuracy</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kelp</td>
<td>663</td>
<td>74.21%</td>
<td>171</td>
</tr>
<tr>
<td>Brown M.A. (except kelp)</td>
<td>34</td>
<td>44.12%</td>
<td>19</td>
</tr>
<tr>
<td>All other M.A.</td>
<td>556</td>
<td>73.38%</td>
<td>148</td>
</tr>
<tr>
<td>Sand</td>
<td>5279</td>
<td>99.98%</td>
<td>1</td>
</tr>
<tr>
<td>All other classes</td>
<td>5780</td>
<td>98.15%</td>
<td>107</td>
</tr>
<tr>
<td>Total</td>
<td>12312</td>
<td>96.38%</td>
<td>446</td>
</tr>
</tbody>
</table>

Table 2.5: Performance on subsets of the data (‘M.A.’ refers to macro-algae), using PCA (RGB) on $15 \times 15$ patches
Table 2.4 shows an analysis of performance on each dive, on the model trained on the full Training Set. It is clear that there is significant performance variation across the dives (note, however, that the f1-score for KELP is misleading on subsets where the true amount of KELP is very small, and undefined with 0% kelp).

Table 2.5 shows a breakdown of where errors occurred in the test set, on various subsets of the data. This table reveals a highly skewed error distribution. Despite comprising nearly half the set, only one sand point was incorrectly labelled as kelp. In contrast, the error rates on the macro algae (particularly the brown group, of similar colour to kelp) were significantly higher. Collectively, the combined macro algae groups contribute 61% of the false positives, despite the fact they constitute only 5% of the OTHER instances in the test set. While it is unsurprising that the model has most difficult with other types of macro algae, the magnitude of the problem should provide a focus for further research.

2.7 Conclusion

In this paper, we have introduced a large data set of expert labelled AUV images, computed benchmark results on a number of different techniques, and examined the behaviour of an automated species detection system in detail. Results on detection of kelp were promising, and the work could be used to develop a practical system to assist marine scientists. For the kelp detection problem, future work that may improve performance includes: combining multiple scales; finding superior descriptors; and making use of the additional non-image information embedded in the AUV data.

For the evaluation of the model, we provided evidence to suggest that a fixed species detection model with good generalisation to at least a local geographic region was
feasible. The skewed error distribution in Table 2.5 highlights the importance of creating detailed performance metrics for end-users, to avoid biasing their scientific outcomes. Finally, the problem should be expanded beyond *Ecklonia Radiata* to allow discrimination between multiple species and features.
Figure 2.5: Percentage distribution of most commonly occurring class labels in each dive mission. The mission names consist of a number (indicating the chronological order), and keywords describing the local area. Key classes are SAND (sand), ECK (kelp) and MATR (biological matrix, typically a range of biological organisms in one area).
Figure 2.6: f1-scores for colour and greyscale descriptors. The dashed line highlights the performance of the peak performing model.
2.7 Conclusion

**Figure 2.7:** Performance on random subsets of training data

**Figure 2.8:** Performance on single dive training sets
Figure 2.9: Performance on discarding individual dives from training set
3 Hierarchical Classification in AUV Imagery

As with Chapter 2, this chapter was presented and published at a conference (Field and Service Robotics 2013 [14], and a poster presentation at the Fine Grained Visual Classification Workshop of Computer Vision and Pattern Recognition 2013), and is included verbatim. Building on the ideas from Chapter 2, the same data set is used; however, the problem is restated from detection of a single species (*Ecklonia radiata*), to recognising a scientifically defined taxonomy of 19 classes. This challenge, of performing supervised learning to produce a hierarchy of labels as the output, is the central idea of the remainder of this thesis. While prior work on hierarchical classification exists, the application to scientific taxonomies in environmental monitoring robotics is novel. A brief review of hierarchical classification in other fields is included here, with a more extensive review in Chapter 5. Where this chapter is focused on re-framing automated benthic image annotation as a hierarchical classification problem, Chapter 5 moves on to a deeper analysis with contributions applicable to the broader field hierarchical classification.

NB: The technique described in this chapter as “Maximum Probability Switching (MPS)” is later referred to as a “Greedy Walk”, to suit the prevalent usage within the field. The original terminology is retained in this chapter to preserve the integrity
Chapter 3 Hierarchical Classification in AUV Imagery

of the published manuscript.

Contribution: I was the primary author of the conference paper, and received support, guidance and other input from the other authors listed in the citation [14]. Dushyant Rao conducted the feature learning component of the work independently. Special acknowledgement is due to Navid Nourani-Vatani, who gave extensive input to the manuscript, and prepared and delivered the conference presentation when I was unable to do so.

Abstract: In recent years, Autonomous Underwater Vehicles (AUVs) have been used extensively to gather imagery and other environmental data for ocean monitoring. Processing of this vast amount of collected imagery to label content is difficult, expensive and time consuming. Because of this, typically only a small subset of images are labelled, and only at a small number of points. In order to make full use of the raw data returned from the AUV, this labelling process needs to be automated. In this work the single species classification problem of [12] is extended to a multi-species classification problem following a taxonomical hierarchy. We demonstrate the application of techniques used in areas such as computer vision, text classification and medical diagnosis to the supervised hierarchical classification of benthic images. After making a comparison to flat multi-class classification, we also discuss critical aspects such as training topology and various prediction and scoring methodologies. An interesting aspect of the presented work is that the ground truth labels are sparse and incomplete, i.e. not all labels go to the leaf node, which brings with it other interesting challenges. We find that the best classification results are obtained using Local Binary Patterns (LBP), training a network of binary classifiers with probabilistic output, and applying “one-vs-rest” classification at each level of the hierarchy for prediction. This work presents a working solution that allows AUV
images to be automatically labelled with the most appropriate node in a hierarchy of 19 biological groupings and morphologies. The result is that the output of the AUV system can include a semantic map using the taxonomy prescribed by marine scientists. This has the potential to not only reduce the manual labelling workload, but also to reduce the current dependence that marine scientists have on extrapolating information from a relatively small number of sparsely labelled points.

3.1 Introduction

Autonomous Underwater Vehicles (AUVs) have made a significant impact on areas of marine science that require an understanding of the sea floor [97, 84, 87]. Common practice is shifting from using human divers with hand-held cameras, to sending AUVs equipped with stereo cameras and other sensors to capture benthic images. These AUVs are capable of capturing far more data, both in the type (various sensor modalities) and volume (a several hour mission can collect tens of thousands of spatially registered stereo image pairs). In addition, the imagery is geo-referenced far more precisely, and data can be gathered from beyond diver depths. While using an AUV improves the raw data in many ways, the sheer volume of it introduces a new problem for interpretation.

For marine scientists studying the location, distribution and coverage of benthic organisms and morphology, the state of the art is to take a small subset of images from an AUV survey, and manually label the content. For the data set used in this paper [5], biological species and physical formations under 50 randomly selected pixels from every 100th image were labelled. The scientists then extrapolate from this subset, to make inferences about the ecosystems and populations in the geographic area. Figure 3.5 shows an example image from our data set (described in Section 3.3)
where 50 randomly selected pixels have been labelled using the CPCe labelling software [58].

![Example AUV image, with CPC point labels](image)

**Figure 3.1:** Example AUV image, with CPC point labels

The ultimate goal of research in this field is to automate the labelling process, such that the vehicle returns a semantic map of the environment, rather than simply providing the raw visual and sensory data. From a robotics perspective, this requires the robotic system to take on the translation and communication effort with which the human users are currently burdened. Recent work [12] describes a solution to a simplified problem of automatically classifying a single species (*Ecklonia Radiata* commonly known as “Kelp”), based on supervised learning algorithms applied to a large AUV dataset. In this paper, we extend the problem to multiple species and physical morphologies, using the same data set.
3.2 Hierarchical Classification

In order to perform classification on a large set of benthic classes, a different approach is required. Supervised machine learning is typically performed on binary problems (such as presence/absence detection) [33, 84], or multi-class problems where there are a set of strict alternatives [9, 66]. In this study, we examine solutions to the more complex problem of predicting a hierarchy of classes, which is the semantic output that is desired by the marine science community that uses this data set (see Section 3.3 for details). The contribution of this paper is to demonstrate and evaluate the techniques used in other areas (computer vision, text classification and medical diagnosis) applied to hierarchical classification in benthic images.

3.2.1 Classification

In a recent survey, Silla Jr and Freitas [81] extensively reviewed the problem of supervised hierarchical classification and defined a range of methods of dealing with the hierarchy of the data. They defined three main approaches:

1. Ignore the hierarchy and perform flat classification (3.2a)

2. Use a network of several local binary classifiers for various regions of the hierarchy (3.2b)

3. Use a single classifier but encode the hierarchical structure somehow in the data (3.2c).

From the list of methodologies presented, we employ the Local Classifier per Node approach of 3.2b. In this approach, which is by far the most used in the literature [81], each node in the classification tree has a binary classifier that is trained to distinguish that class from others. This approach is relatively straightforward to implement, but has several advantages:
Figure 3.2: Classification approaches for a hierarchical data set. The dashed lines show the extent of each classifier. In the flat (3.2a) and global (3.2c) approaches a single classifier is employed whereas in the local approach (3.2b) several classifiers are used; the shown tree is the Local Classifier per Node approach. Figures taken from [81].

- It is possible to either output the single node in the hierarchy which represents the best prediction, or to query the probability of a given image point being a member of any given node.

- It allows use of the full data set, including the points which have not been labelled to a leaf node (e.g. as simply “biological”, or “algae” rather than specific species).

- It allows different features, classification algorithms and even training sets to be used for each classification sub-problem. Intuitively, it makes sense that the features which distinguish between various species of algae would differ from those that distinguish sand from rock.

3.2.2 Training policies

In training a network of binary classifiers, it is necessary to decide which examples to use as the training set for each node, and how feature extraction is performed at each node. We compare the two policies described in [81] that most naturally fit our problem: the inclusive and the sibling policies.
3.2 Hierarchical Classification

3.2.2.1 The inclusive policy

This policy includes as positive examples the entire subtree of the training node. The nodes in the rest of the tree are used as negative examples, with the exception of direct ancestors of the training node. The direct ancestors cannot be used as either negative or positive examples as they contain instances of both; e.g. if training a classifier for node 2.1 in 3.2b, then nodes 2.1.1 and 2.1.2 are used as positive examples and all other nodes with the exception of the ancestor nodes R and 2 are used as negative example (nodes 1, 1.1, 1.2, 2.2, 2.2.1 and 2.2.2).

3.2.2.2 The sibling policy

This alternative policy uses the same positive training examples, however, the negative examples are restricted to siblings of the training node (and not siblings of the ancestor nodes). Using the example from before, the negative training samples will come from nodes 2.2, 2.2.1 and 2.2.2 only.

The expected performance difference between these two policies is not obvious, with no consistent winner found in [40]. On one hand, the inclusive policy ensures that each node is as informed as possible, and should be able to deal better with classifying instances that belong elsewhere in the tree (as it has seen examples of all the data). On the other hand, the siblings policy solves a much more specific problem (distinguishing a node’s class from its siblings), and may give better discriminative performance between these classes. Such nodes will, however, be less informed about instances that belong elsewhere in the tree. An inherent advantage of the siblings approach is that far less training data is required for nodes deeper in the tree, which becomes significant when the tree is large.
3.2.3 Prediction

After training the network of classifiers, the decision remains of how to predict the class of a given image point. As we require complete consistency in the hierarchical labels (such that there is a single, unbroken chain of classifiers predicting positive results from the root to the deepest node), the simplest choice described in [81] is chosen, which we refer to as maximum probability switching (MPS). An instance starts at the root node, and flows to the leaf node with the highest prediction probability (akin to performing one-vs-rest classification at each node). This technique forces prediction down to the leaf node level. We can remove this constraint by stopping a prediction from moving further down the tree when the maximum predicted probability falls below some threshold (say 0.5 for the typical cut-off of a binary classifier).

We also test an alternative approach: the use of a simple probabilistic graphical model (PGM). Here the class tree also represents the independence relations in the PGM, and we assume the conditional probability of node membership is given by the probabilistic predictions of the classifiers\(^1\). This allows exact Bayesian inference to be performed trivially, by multiplying probabilities of a node’s ancestors to obtain the net probability of membership.

3.2.4 Performance measure

Lastly, a robust performance metric is needed; ideally a single number to evaluate the performance on an entire tree. The closest performance measure commonly used in the literature is the hierarchical f1-score [56]. Each instance has multiple counts

\(^1\)This assumption is at least reasonable for the siblings training policy, as the training sets only include those instances which we know belong to the parent node. The output of the classifiers therefore represents e.g. “the probability that this instance is Algae, given we know it is an instance of Biota".
of true/false positives/negatives, as each node in the chain of true class nodes is compared to the chain of predicted nodes. 3.3a and 3.3b illustrate two scenarios. In the former case, the prediction down the entire tree is accurate. In the latter case the last prediction is incorrect. This results in a False Negative (FN) detection as the correct node was missed as well as a False Positive (FP) detection for an incorrect node being detected.

When employing thresholded MPS it is possible that the prediction stops earlier up in the tree than the ground truth. This occurs when the child classifier prediction probability is below the set threshold. In such an instance, the missed nodes are calculated as FNs. This is illustrated in 3.3c.

Finally, there can be cases where the predicted class is more specific (lower down the tree) than the ground truth class. This can occur when the ground truth label is not at the leaf level. In such an instance the hierarchical f1-score commonly used in the literature would penalise the prediction with a FP. However, there are instances where this is not fair; e.g. the classifier predicts “Labrador” while the label says “Dog”. “Labrador” is of course a breed of “Dog”, and may be a more accurate description. This is true for our labels of the underwater species, where the labeller does not always label the instance to the leaf-node level. This can occur due to poor image quality, inability of the labeller to recognise the species or cost and effort constraints if more detailed labels are not required for the given research project.

We therefore modify the metric such that if the predicted class is more specific than the ground truth (3.3d), we do not reward or penalise any results deeper than the deepest known class.
(a) True prediction: (TP, TP, TP)
(b) False prediction: (TP, TP, FP, FN)
(c) False prediction due to thresholding: (TP, TP, FN)
(d) Modified prediction: (TP, TP)

**Figure 3.3:** The hierarchical f1-score performance metric. The connected circles represent classifier nodes, with the root on the left, and leaf on the right. Grey circles are ground truth, white circles are prediction. 3.3a shows the case of the classifier network correctly predicting a node of depth 3 in the hierarchy. 3.3b represents the first two levels being predicted correctly, whereas the classifier network incorrectly chooses a sibling of the true class at depth 3.

### 3.2.5 Descriptors

A subtlety of the Local Classifier per Node approach is in the selection of image features. In flat multi-class classification (3.2a) and global classification (3.2c), the same image features are typically used for the nodes. With the local classifiers (3.2b), we can select (either manually, or by feature learning techniques) features that are optimised per classifier node for the siblings or inclusive training data sets. In this paper we employ various features that both optimise for the local node or are constant across the nodes.

We analyse the performance of Principle Component Analysis (PCA), Local Binary Patterns (LBP) texture descriptors, and feature learning (FL). In all instances, the features are extracted from localised image patches around the ground truth label. As discussed in [12], the AUV travels at an approximately constant height of 2m above the sea floor. We therefore use the same assumption of pixels in the image representing a fixed scale. In the $1360 \times 1024$ RGB images, 100 pixels corresponds to approximately 10 centimetres.
3.3 Data set and Hierarchy

3.2.5.1 Principal Component Analysis

The PCA descriptors are calculated on each of 7, 15, 31, 63 and 95 pixel square RGB image patches. The patches are whitened and empirical study showed that keeping 60 components is sufficient.

3.2.5.2 Local Binary Patterns

We investigate two varieties of the popular LBP descriptors, namely the rotation-invariant uniform descriptor [70] and the histogram Fourier transform descriptor [1]. The descriptors are calculated from a $31 \times 31$ pixel patch and normalised using various image normalisation methods.

3.2.5.3 Feature Learning

Feature learning uses K-means clustering to learn a dictionary of 1000 patches for both 7 and 15 pixel square images. The CIFAR-10 Images Dataset [60] was used to generate the features, instead of the Tasmanian AUV image data. Compared to the AUV images, the CIFAR-10 data set has more variation in the content, and is therefore able to produce a more diverse dictionary of learned features. The features are encoded using the L2 distance to each patch in the dictionary [27]. The centroids, or learned features, tend to resemble Gabor-like edge or texture filters, as shown in Figure 3.4.

3.3 Data set and Hierarchy

The Tasmania Coral Point Count data set is comprised of 14 separate dive missions conducted by the AUV *Sirius* off the South-East coast of Tasmania, Australia, in
October 2008. From the data set containing over 100,000 stereo pairs of images, marine scientists at the University of Tasmania [5] selected every 100\textsuperscript{th} colour image and used the CPCe software package [58] to label 50 random points on each [5]. This has resulted in 62,900 labels from 1258 images. Figure3.5 illustrates some images from the data set\textsuperscript{2}, with original and predicted labels.

A wide range of class labels were used, indicating biological species (including types of sponge, coral, algae etc.), abiotic elements (types of sand, gravel, rock, shells etc.), and types of unknown data (ambiguous species, poor image quality, etc.). Precise details of the labelling methodology can be found in [5].

Recently, a standardised tree structure for biological and physical classes for underwater species was defined as part of the Catami project [2]. By mapping the data set to this classification hierarchy, a complex tree of 19 classes was obtained (See Figure3.7). Note that we have merged some of the labels under their ancestor nodes. These labels corresponded to species with very few instances, because they belonged to an unknown species or a mixture class. Despite this consolidation, the class instances remain highly unbalanced; e.g. the “SOFT” class has 6139 instances while the “BROWN” erect-branching algae has only 10 instances.

\textsuperscript{2}The data set is available from \url{http://marine.acfr.usyd.edu.au/datasets}
3.4 Results and Discussion

3.4.1 Hierarchical Classification Approaches

We present results using logistic regression (LR) classifiers, with features derived using PCA, LBP texture features, and feature learning (FL), and compare the *sibling* and *inclusive* policies.

Performance is measured in terms of the modified hierarchical f1-score with the same
Figure 3.6: Comparison of *Siblings* and *Inclusive* policies using the modified hierarchical f1-score. Each marker represents both *sibling* and *inclusive* performance on a given feature and prediction setup. Marker size is scaled by the image patch size used.

In Figure 3.6, performance using the modified hierarchical f1-score is compared for the two policies across a range of image descriptors and prediction approaches. As can be observed, there is a clear trend that for mandatory leaf node prediction, the PGM is generally superior to MPS. This is promising for future work, as the PGM is the more principled approach, and more sophisticated models can be used. Also, the modification to permit the network to predict only higher level classes when less confident (using thresholding) was highly successful at improving the performance.

In terms of *inclusive* and *sibling* policies, we obtain the same finding on underwater
3.4 Results and Discussion

images as that found using text classification [40]—no clear winner. Given the sibling policy has a significant advantage in reducing training time, it is preferred in situations where the results are comparable.

The best result was accomplished using the LBP descriptor, the inclusive training methodology and thresholded MPS prediction with a threshold of 0.5. This resulted in a tree f1-score of 80.2%. The detailed performances are shown in Figure 3.7. We notice that the performance at the highest level of the hierarchy (differentiating between biological and physical) is in the mid 80’s. The result is also satisfactory, compared to [12]\(^\text{3}\). However, for nodes without a significant number of training instances, performance was very poor.

3.4.2 Hierarchical vs flat multi-class classification

In order to compare the approach of the hierarchical binary network and traditional flat multi-class classification, it was necessary to restructure the problem. For the hierarchical approach, MPS was used to force predictions to leaf node level. The flat classifier was comprised of the 13 leaf nodes in the tree (selected as per 3.2a). The test set was reduced to contain only data that had been labelled to leaf node (as higher level labels have no defined assignment in a leaf node flat classifier). Rather than the tree f1-score (which also makes little sense for a flat classifier), results were evaluated using the mean f1 score across the leaf node classifiers. As the mean f1-score weights nodes equally, the overall results were significantly lower than the tree f1-score. On this metric, the hierarchical classifier’s performance was marginally higher. This was largely due to superior performance on the most poorly performing nodes (e.g. the f1-score on Cnidaria reduced from 0.09 to 0.00 with the flat multi-class case). Note that this comparison was performed using the siblings

\(^{3}\text{i.e. in the ability to identify }\text{Ecklonia radiata}\)
Figure 3.7: Performance results on best classifier (HLNP with thresholding, LBP features, inclusive training set). The grid of 4 numbers is the confusion matrix for instances in the test set, which was used to compute both the local f1-scores for each node (red bars), and the tree f1-score (given in the RootNode box, and calculated by computing the f1-score on the sum over the local node confusion matrices). The thickness of the grey edges between nodes is proportional to the number of instances from the test set that belong to a given node.
3.5 Conclusion

We have investigated various aspects of performing supervised hierarchical classification on sparsely labelled benthic imagery for the purpose of species recognition. The aim was to apply techniques used in the literature from other fields, to construct an initial solution for the automated interpretation of AUV images.

Results have shown over a range of feature descriptors and patch sizes, that with the PGM prediction, better results were obtained than using simple MPS. This is promising for future directions as it is a more principled approach. However, the best results were achieved employing thresholded MPS.

We also compared two different classifier training approaches, namely the local sibling policy against the global inclusive policy. It was demonstrated that comparatively, the sibling and inclusive training policies exhibit similar performance, with the sibling option holding the advantage due to reduced training time.

In addition, the comparison with a flat multi-class approach on the reduced data set confirms that basic performance at leaf-node level is at least as good as the traditional approach, using our hierarchical classification scheme.

Future work will cover a number of areas. In terms of the hierarchical classification, we will investigate potential improvements, such as the use of PGMs with variable

<table>
<thead>
<tr>
<th></th>
<th>Mean f1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical (trained on leaf-node training data)</td>
<td>0.197</td>
</tr>
<tr>
<td>Hierarchical (trained on all training data)</td>
<td>0.182</td>
</tr>
<tr>
<td>Flat multi-class (trained on leaf-node training data)</td>
<td>0.178</td>
</tr>
</tbody>
</table>

Table 3.1: Flat multi-class and hierarchical classification comparison, tested on the reduced leaf-node only data set
depth prediction. Because the classification scheme permits different features to be used at different nodes, another challenging area of research will be to find ways of incorporating other sensor modalities from the AUV (such as dense stereo information) to enhance the ability of the classifier to distinguish between various species and objects. Although the automated semantic labelling described in this paper has been applied as a post-processing step, eventual incorporation as a real time algorithm on-board the robot would have further benefits. Communication with the AUV through water is typically performed using an acoustic modem, which has far lower bandwidth than what is necessary to transmit raw image and sensor data. If the robot can “understand” what it sees by assigning automated labels, it could use that information to either adapt its behaviour, or relay it to the human operators for monitoring and intervention.
4 BENTHOZ-2015: An Australian benthic data set

The automated semantic interpretation of AUV images relies on research from various fields (e.g. machine learning and computer vision) being applied to work done in marine robotics and marine science. A fundamental limitation to this is open data. Without open data sets, every piece of work seeking to make progress in the field is necessarily done by researchers with close links to the robotics and marine science groups that captured and annotated the data. For those able to access relevant data, it is typically necessary to manually define and process a usable data set. As well as requiring a substantial investment of time, it is then difficult to make comparisons with work done by other researchers, or for others to replicate or build on results.

The solution to this problem in the computer vision and machine learning communities has been to create open data sets, on which researchers publish the performance of their algorithms. In this chapter, we have assembled and published a comprehensive data set containing AUV captured benthic images, sensor data, and expert annotated labels from around Australia. This work required collaboration between five research institutions, and a large number of contributors (as evidenced by the author list). We use it as a consistent data set for work in the remainder of this thesis, and invite other researchers to use the public data set to reproduce our results,
improve on our algorithms, or conduct their own research.

The text below was published in Nature’s Scientific Data journal as a data set descriptor in October 2015 [13]. Both the paper and data set were released under the Creative Commons 4.0 license, in accordance with the Nature Publishing Group’s policies.

Contribution: I organised the collaboration, prepared the data set from the files provided to me by the marine scientists, and wrote the majority of the manuscript. Marine scientists wrote and submitted their own sections describing methodology and validation methods in their geographic region (organised by Renata Ferrari). Authors listed on the publication were: Michael Bewley, Dr. Ariell Friedman, Dr. Renata Ferrari, Dr. Nicole Hill, Dr. Renae Hovey, Dr. Neville Barrett, Dr. Oscar Pizarro, Dr. Ezequiel Marzinelli, Dr. Will Figueira, Ms Lisa Meyer, Russell Babcock, Dr. Lynda Bellchambers, Prof. Maria Byrne, Prof. Stefan Williams. NB: Dr. Marzinelli was added since the original version as an erratum.

Abstract: This Australian benthic data set (BENTHOZ-2015) consists of an expert-annotated set of georeferenced benthic images and associated sensor data, captured by an autonomous underwater vehicle (AUV) around Australia. This type of data is of interest to marine scientists studying benthic habitats and organisms. AUVs collect georeferenced images over an area with consistent illumination and altitude, and make it possible to generate broad scale, photo-realistic 3D maps. Marine scientists then typically spend several minutes on each of thousands of images, labeling substratum type and biota at a subset of points. Labels from four Australian research groups were combined using the CATAMI classification scheme, a hierarchical classification scheme based on taxonomy and morphology for scoring marine imagery. This data set consists of 401,850 expert labeled points from around the Australian
coast, with associated images, geolocation and other sensor data. The robotic sur-
veys that collected this data form part of Australia’s Integrated Marine Observing
System (IMOS) ongoing benthic monitoring program. There is reuse potential in
marine science, robotics, and computer vision research.

4.1 Background & Summary

Less than 0.05% of the global sea floor has been mapped with sonar swath mapping [28] at high resolution (tens of meters). Coverage at visual resolution (millime-
ters) using a camera is substantially lower. Visual resolution permits the detailed
analysis of benthic taxonomy; however, this requires image capture at an altitude
of several meters above the sea floor, typically traveling slower than walking pace.
The growing maturity of AUVs has permitted broader and more systematic visual
surveys than traditional diver held cameras or towed video sleds (a system whereby
a camera on an underwater sled is attached to a ship by a cable, and towed. The
resulting images are lower quality than an AUV as the positioning, particularly alti-
tude, is difficult to control precisely). AUVs can operate continuously and precisely
at greater depths, with geolocation, sensor data and stereo images captured several
times a second. A 3D visual map of the survey area can then be produced from the
data. This abundance of data has introduced a new problem for scientists: efficiently
extracting and distilling useful information from the raw data.

The data set presented in this paper contains 401,850 expert annotations of 9,791
georeferenced images with associated sensor data (latitude, longitude, depth, alti-
tude, salinity and temperature) from around the Australian coast (see Figure 4.1).
The annotations conform to a hierarchy of 148 substratum and biological classes
(Figure 4.2), and specify the content at specific points within each image. All image
and sensor data were captured by the *Sirius* AUV. *Sirius* is the primary platform responsible for collecting seafloor images as part of the AUV facility of the Integrated Marine Observing System (IMOS) in Australia [37]. Table 4.1 summarizes the number of expert labels applied to each campaign, and Figure 4.1 shows the geographic location of each deployment. The annotation process poses a significant bottleneck, taking a trained marine scientist 5 minutes or more to assign semantic labels to dozens of individual points on a single image using the context provided by the image neighborhood around the point. After a survey is conducted, there is typically a time lag of several years before the labeling is complete, and scientific inferences can begin to be drawn. Even with this delay, it is only practical to label a very small fraction of the data collected by the AUV. For the deployments in this data set, the 9,791 images with labels represent around 2% of the total number of images captured during those deployments.

Machine learning and computer vision techniques have the potential to increase the amount of labeled data and reduce the time it takes to do so. The availability of a set of high quality expert labels with geographic and temporal diversity will permit researchers in these fields to investigate ways to reduce or eliminate the manual labeling effort, as well as gaining new scientific insights from working with a combined data set. Another significant hurdle to the integrated analysis of benthic imagery data is the lack of standardization between research groups. Until recently, individual research groups have labeled images using a variety of custom labeling systems and standards suited to their particular geographic region and research interests, which limits the ability to perform scientific analysis, or train machine learning algorithms on large, varied data sets. In this data set, however, we combine data from four leading research groups, using the recently established Collaborative and Automated Tools for Analysis of Marine Imagery (CATAMI) class hierarchy [2] as
a standardized labeling scheme. The CATAMI scheme permits the various schemes to be combined in a consistent and meaningful way, as shown in Figure 4.2.

![Location of AUV Campaigns](image)

**Figure 4.1:** Geographic distribution of annotated images

### 4.2 Methods

#### 4.2.1 AUV Data Collection

This data set includes annotated images from an extensive series of AUV-based benthic surveys that were undertaken between 2008 and 2013, around Australia’s coastline. Geographic locations include Western Australia, Tasmania, New South Wales and Queensland (Figure 4.1). Image and sensor data was gathered by the AUV *Sirius*, as described in [97]. The campaigns were conducted by the AUV facility of
the IMOS program, funded by the Commonwealth Government and collaborating agencies (see Acknowledgements). During each campaign, *Sirius* executed several missions, deployed at sites selected by the science party, typically focusing on temperate rocky reefs and coral reefs. Missions are defined by a set of georeferenced waypoints and instructions. The AUV autonomously captured images every 0.5 seconds, while maintaining a nominal 2 m altitude above the sea floor.

The general sampling methodology is described in [97] as being designed to monitor the fundamental reef processes that maintain reef biodiversity and resilience. The processes of interest occur at a number of spatial scales, so a nested hierarchical sampling design was adopted to allow changes to be observed at the differing scales. Deployment mission designs included: 1) Long transects used to monitor broad community structure and integrity, community boundaries, and transitions 2) broad scale, sparse grids on the order of 500–1,000 m on a side to determine spatial variability in habitat structure 3) small-scale 25 m × 25 m full-cover dense grids, providing contiguous coverage mapping for the establishment of long-term monitoring sites. Further detail on target habitat locations, overlapping survey patterns, and other aspects of survey design can be found in [97]. Deployments were performed from a ship and typically lasted a few hours, resulting in tens of thousands
of stereo image pairs and associated sensor data per dive.

4.2 Methods

4.2.2 Expert Annotations

The general approach to annotating images across the four research groups was the same. A subset of images from the dives were selected (e.g. every 100th image), and the commonly used software package Coral Point Count with Excel Extensions (CPCe) [58] was used to label the content beneath up to 50 uniformly randomly selected points within the image (where the label represents the content under that point, rather than a larger area around it). It should be noted that the data set is therefore unsuited to estimate abundance of rare classes based on individual images; the intended use is to compute statistics over a 25 m x 25 m or larger area, or along a transect. Further discussion of marine science literature using up to 50 points per image can be found in the final section of this descriptor.

For the purpose of this data set, the labeling schemes used by the individual research groups were mapped onto the CATAMI hierarchy [2] (see Figure 4.3 for an example). Mapping files used to convert original CPC codes to CATAMI classes were reviewed by the respective marine science groups, and are included as supplementary files (Data Citation 1), along with a description of each class in the CATAMI hierarchy (Data Citation 1). The following sections describe the differences in methodology between the research groups.

4.2.2.1 Western Australia

Images were obtained from three key locations along the Western Australian coastline; Rottnest Island, Jurien Bay and the Houtman Abrolhos Islands. Rocky reef/ coral habitat was targeted to establish a series of reference sites for long-term monitoring
in the west coast bioregion. Sites were selected based on bathymetry maps and existing knowledge, to target moderate to high relief reef between 15 and 30 m depth. Within each site, 3 replicate grids were surveyed by the AUV which was achieved by conducting a series of parallel, overlapping 25 m long transects, covering a combined area of 625 m$^2$ of seabed (i.e. $25 \times 25$ m). Replicate grids within a site were positioned approximately 200 m apart. Over 1,000 georeferenced stereo image pairs were collected from each grid. These high resolution images were subsampled at 20 second intervals to generate a sample set of 101-129 non-overlapping images that maximised spatial coverage of each grid [82]. For image analysis, 50 random points were digitally overlaid onto each sample, and the number of points covering each benthic category was counted (using CPCe [58]), then doubled to give a proxy of percent cover. Forty benthic categories, including dominant flora, fauna and substratum characteristics, were determined a priori based on previous research [83] and used to classify each image. Care was taken to include conspicuous species of considerable ecological importance, such as the canopy-forming brown algae *Ecklonia radiata* and *Scytothalia dorycarpa*, while also using functional or morphological groups to achieve a broad, holistic approach to describing the benthos. Bleached coral was considered “alive”, but additional information on the spatial extent of any bleaching was recorded.

### 4.2.2.2 New South Wales

Images obtained during AUV surveys were used to quantify benthic assemblage structure and composition on rocky reefs at three locations along the NSW eastern coastline (Figure 4.1). These three locations included highly diverse sub-tropical and temperate rocky reefs between 20 and 50 m depth [63]. Within each location multiple 625 m$^2$ dense grids of rocky reef were surveyed at multiple sites using the *Sirius* AUV,
4.2 Methods

sites were at least 1km apart. The AUV achieved full coverage of each 625 m$^2$ dense grid (15,000 image pairs), from which 50 spatially balanced images were selected using a generalized random tessellated stratified design in R package \textit{spsurvey} [55]. Each image covered an area of approximately 1.8 m$^2$; so, 50 images covered 15\% of a 6,625 m$^2$ dense grid. Twenty five random points were overlaid on each image and taxa under each point were identified to the highest taxonomic resolution possible using CPCe. The national standard classification scheme CATAMI Version 1.2 was used to identify organisms to a taxonomic, morpho-group (e.g. encrusting coral), major group (Class) and / or morphological level [2].

4.2.2.3 Tasmania

AUV campaigns conducted in 2008 and 2009 targeted reef systems on the Tasman Peninsula on the South-East Coast of Tasmania. AUV transects followed an elongated grid design where the ‘long’ section of transects was oriented down the depth gradient and ‘short’ sections of transect were oriented across the depth gradient. Every 100$^{th}$ image along the transect path (a spacing of approximately 40 m) from one camera was scored using CPCe. In preliminary analyses on the Tasman Peninsula, a spacing of 40 m meant that images generally occurred in the next patch of substratum along the transect and the range of substrata and the values of multi-beam derived variables sampled in images was representative of that found in the entire study region. The substratum or biota underneath 50 random points within an image was scored. Benthos was identified to the lowest possible taxonomic or morphological unit using [39] and [47]. For most sessile invertebrates this was morphospecies, identified by morphology and color. Representative algae were identified to species, otherwise to functional groups, and mobile invertebrates (infrequently observed) were assigned to broad categories (e.g. starfish, sea urchin, mollusc). This
scoring approach pre-dates the CATAMI classification scheme, and contains a number of highly specific classes (at species level). For the purpose of this data set, labels were mapped post hoc to the scheme, where the scored class was matched to the deepest valid level of the hierarchy.

4.2.2.4 Queensland

The AUV campaign conducted in Queensland in 2010 focused on reef systems east of Moreton Island in southeast Queensland. The AUV mission was intended to cover the full depth range of the reef at Henderson’s South which was approximately 12 m in depth at its shallowest to over 45 m at its greatest depth. A gradation in habitat types was known to occur at this site, transitioning from turf algae and corals in the shallower parts of the reef to kelp forest in the deeper areas. For this reason a regular sub-sampling was undertaken, allowing for the mapping of the spatial features of habitat structure. Transects were located between the depths of 17-42 m on a 400x500 m-rectangular grid design with intersecting lines spaced every 100 m. The grid was oriented so that there were five lines roughly E-W perpendicular to depth contours and six N-S roughly parallel to depth contours. From each of these lines 100 m transects were sub-sampled, along the side of each of the cells outlined by the square grid. Ten images were selected from each transect at a spacing of 10 m. The substratum or biota underneath 20 random points within an image was scored using CPCe. Benthic biota were identified to the lowest possible taxonomic or morphological unit using the CATAMI classification scheme. For most sessile invertebrates this was morphospecies, differentiated by morphology and color. Representative algae were identified to species, otherwise to functional groups, and mobile invertebrates (infrequently observed) were assigned to broad categories (e.g. starfish, sea urchin, mollusc).
4.3 Data Records

4.2.3 Code availability

The production and processing of this data relied on a complex software pipeline, involving controlling a hover-cable AUV, extracting and post-processing the data to produce accurate georeferencing via a Simultaneous Localization and Mapping (SLAM) algorithm, using CPCe to annotate images, and further scripts to import data into the Squidle benthic imagery web application. The code for Squidle, along with the import scripts, are available on github at github.com/acfrmarine/squidle. The script used to import the cpc files into the squidle database is located in that repository at scripts/annotation-scripts/import_cpc_file.py. The most complete descriptions of the data acquisition process are in [54] for image processing, and [62] for navigation and SLAM.

4.3 Data Records

The complete data set described here has been made available on Squidle at http://squidle.acfr.usyd.edu.au. Squidle is a new web-based framework that facilitates the exploration, management and annotation of marine imagery. It provides a user-friendly interface that integrates spatial map-based data management tools with an advanced annotation system. The online annotation system permits scientists to easily collaborate on both the labeling and use of their data. It will in future also provide a platform for using and testing machine learning and computer vision algorithms on marine imagery. This data set has been made available to view, explore and download via the web interface. Most of the expert annotations were produced prior to the development of Squidle and the CATAMI scheme, and have been imported into Squidle CPCe. By comparison, CPCe is a standalone application for individual users to label marine images locally, and does not include higher level
features such as data exploration and online collaboration. A comparison between
Squidle, CPCe, and other platforms was performed in [3].

It should be noted that the although the Squidle platform is the easiest way to
derive the data set, it is still under active development. An image downloader tool
is available on the site, as well as the ability to download non-image data as csv
files. In addition to Squidle, the data set is also available from a number of other
sources.

A table of expert annotations, sensor data, geolocation and image metadata for
BENTHOZ-2015 is available for download from figshare (Data Citation 1), the online
scientific data repository.

All images and sensor data (without expert annotations) captured by the AUV Sirius
IMOS AUV survey data; as such it contains images and sensor data from a large
number of surveys not included in this data set. Note that there is no mechanism
to select the precise set of data defined by BENTHOZ-2015.

4.3.1 Expert Labels

The expert labels are available as a comma separated value file (Data Citation 1),
where each row represents a single expert labeled point within an image. The fields
are described in Table 4.2, and consist of a unique identifier, the image containing
the point, and the location of the point within that image.

The “label” field defines the class within the CATAMI hierarchy that has been
assigned to each point. Figure 4.4 shows the frequency of the most popular class
4.3 Data Records

labels appearing in each region, after mapping to the CATAMI hierarchy. Note the heterogeneity in the labeling with, for example, *SUS* (sandy substrate) dominating in Tasmania, and *MA* (Macro Algae) dominating in Western Australia. These differences should not be solely attributed to differences in biogeography, due to the deliberate non-random selection of deployment sites with different scientific aims (the only true random samples in the data set are the selection of N points within each annotated image). Each label in the hierarchy corresponds to a *Codes for Australian Aquatic Biota* (CAAB) Code, which acts as an Australia-wide identifier for aquatic organisms. The codes are described in more detail at [http://www.marine.csiro.au/caab/](http://www.marine.csiro.au/caab/).

It is also important to note that some of this heterogeneity is lower than initially appears - e.g. *MAENR* (Macro Algae Encrusting Red) is a more specific type of *MA*. This variation in specificity of labels is driven by research groups’ areas of expertise, and scientific interests.

<table>
<thead>
<tr>
<th>Data field</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>kpid</em></td>
<td>A unique identifier for an expert labeled point in an image</td>
<td>string</td>
</tr>
<tr>
<td><em>image_id</em></td>
<td>A unique identifier for the image this point applies to</td>
<td>integer</td>
</tr>
<tr>
<td><em>y</em></td>
<td>Fraction of the point from the top of the image</td>
<td>numeric (0-1)</td>
</tr>
<tr>
<td><em>x</em></td>
<td>Fraction of the point position from the left of the image</td>
<td>numeric (0-1)</td>
</tr>
<tr>
<td><em>label</em></td>
<td>A unique number assigned to the point</td>
<td>integer</td>
</tr>
<tr>
<td><em>code</em></td>
<td>An abbreviation of the class name assigned to the point</td>
<td>string</td>
</tr>
</tbody>
</table>

*Table 4.2: Expert label fields*
4.3.2 Image Metadata

As the AUV captures an image at a particular point in time, other metadata can be assigned to that image, such as vehicle position, and additional sensor data being recorded by the AUV. These fields are described in Table 4.3. Figure 4.5 and Figure 4.6 show the time and depth distributions over which the annotated images were captured in each region. The precise georeferencing represents a significant advantage of using an AUV based data set, aside from the sheer volume of data collected. Repeat surveys of the same areas (some of which are present in this data set) can be compared to accurately evaluate changes over time [22] [20].

<table>
<thead>
<tr>
<th>Data field</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_name</td>
<td>The unique identifier of an image (1360x1024 pixels, RGB)</td>
<td>string (no file extension)</td>
</tr>
<tr>
<td>date_time</td>
<td>Time stamp of image, in UTC</td>
<td>string (YYYY-MM-DD HH:mm:ss+00:00)</td>
</tr>
<tr>
<td>campaign</td>
<td>The campaign during which the image was captured</td>
<td>string (&lt;region&gt; YYYY)</td>
</tr>
<tr>
<td>deployment</td>
<td>The name of the deployment, within the campaign</td>
<td>string (rYYYYYMMDD_HHmmss_&lt;name&gt;)</td>
</tr>
<tr>
<td>latitude</td>
<td>The latitude of the vehicle</td>
<td>Decimal degrees</td>
</tr>
<tr>
<td>longitude</td>
<td>The longitude of the vehicle</td>
<td>Decimal degrees</td>
</tr>
<tr>
<td>depth</td>
<td>The depth from the surface in metres of the camera</td>
<td>Positive numeric (More positive is deeper, in metres)</td>
</tr>
<tr>
<td>altitude</td>
<td>The height of the camera above the sea floor, according to the Doppler Velocity Log</td>
<td>Positive numeric (m)</td>
</tr>
<tr>
<td>salinity</td>
<td>The salinity measured by the vehicle</td>
<td>Numeric (psu)</td>
</tr>
<tr>
<td>temperature</td>
<td>The temperature of the water measured by the vehicle</td>
<td>Numeric (Celsius)</td>
</tr>
</tbody>
</table>

Table 4.3: Image metadata fields
4.3 Data Records

4.3.3 Images

Images are downloadable as PNG files with lossless compression, typically between 1.3 and 2.4 MB per image. Some simple batch processing was performed to enhance the images, described in [54]. Because of the large size of the data set (approximately 10,000 annotated images, or two orders of magnitude larger including unlabeled ones), images are made available using a separate downloader tool. Users are requested to download only the data they intend to use. Squidle and the IMOS AODN Portal also permit exploration of the images using a web browser without requiring a bulk download. Image acquisition is described in detail in [54], using a stereo camera pair. The color camera was used to capture the images used here, which has an approximate field of view of $42 \times 34$ degrees, for the $1,360 \times 1,024$ pixel RGB images. At a typical altitude of 2 m, this corresponds to an image approximately 1.5 m by 1.2 m, with an area of approximately 1.8 m$^2$, and pixels representing approximately 1 mm in extent. The non-flat nature of the sea floor, and camera geometry, mean that these measurements should not be considered precise. The altitude of each image is provided in the data set, so some spatial scaling can be made. The roll, pitch and yaw of the camera is approximately fixed (pointing downward), due to the passively stable design of the AUV Sirius.

4.3.4 Data set definitions

In order to support the needs of machine learning researchers, a separate “secret” test set has been reserved (not included in the numbers and figures in this paper) as (Data Citation 1). This allows predictive models to be developed using the publicly available training data set described here, and then tested against a previously unseen set of test data. The test data set consists of a small number of individual
deployments (representing particular geographic locations) that were selected across geographic location and year. All labeled points on those images are available, but listed as “Unknown” class in the publicly downloadable data. The training set and test set are both available separately for download from Squidle and figshare.

The training set has been organized into the following hierarchy:

- A campaign consists of a series of deployments conducted on a single field trip in a geographic area (e.g. Tasmania 2008)
- Each campaign is broken into a series of deployments, each of which represents a continuous set of data starting when Sirius was launched from the support vessel, until Sirius was recovered, usually several hours later

### 4.4 Technical Validation

Annotation data in this data set were produced by a combination of experienced scientists and trained research students, all scorers having either considerable prior experience in benthic image annotation or extensive training in both benthic image annotation and the used classification scheme. Selection of interested students was undertaken before training so that only the most committed and skillful students with suitable underwater experience were invited to participate. All students underwent the same training, with individual supervision and help with taxa identification. Training lasted between 1 and 4 weeks depending on the student skill and previous experience. Annotations generated during training were not added to the database. The authors trained all non-experts who contributed data to this data set. Quality control of labels varied slightly for each state, as described below. The typical usage of expert annotations in the literature does not take into account a quantification of the label accuracy. There is no practical means by which to obtain
a "gold standard" reference (as the biota and physical morphology changes over time, and precise georeferencing is difficult. Instead, the most suitable means is to compare inter-expert agreement on point labels based on the same set of images. With a data set of this complexity (148 classes, and approximately 10,000 images from a diversity of geographic regions), this validation is best designed and performed with a particular application in mind, and does not require any information other than that contained in this data set and descriptor. Researchers making use of this data set are invited to assess the validity of the data set for their purposes.

4.4.1 Western Australia

A total of 7 people labeled this data set, from which 3 were experts and 4 trained non-expert researchers. Consistency among the 7 labelers was verified by having everyone score the same points in the same images. Emphasis was placed on correct identification of broad groups (algae, coral, sponges, other sessile invertebrates and seagrass), and accurate identification of algal and coral morphotypes (i.e. fleshy vs calcareous, red vs brown or green, plating vs massive coral). A photo identification guide was created and constantly updated and re-circulated to all labelers. Any uncertain labels were flagged and verified by one of the 3 experts. When comparing expert labeled points to trained labeled points, the only difference was that experts tended to label shadowed points with taxa, while trained labelers would classify those as points as uncertain, and redirect them to experts.

4.4.2 New South Wales

A total of 5 people labeled this data set, from which one was an expert and 4 trained non-expert research students. Consistency among the 5 labelers were verified by
having everyone score the same points in the same images. Annotations between trainer and students were carefully compared and students only started annotating after consistency was at least 85% at the morpho-taxa level for all biotic and abiotic groups (e.g. fine branching fleshy red algae). This threshold was typically achieved after 2 – 3 weeks of intensive training. A photo identification guide was created and constantly updated and re-circulated to all labelers. Any uncertain labels were flagged and verified by the expert.

4.4.3 Tasmania

A total of 3 experts labeled this data set, from which 1 coordinated an image catalog and morpho-species names to avoid duplication. Initially, the coordinator scanned through the imagery and identified and cataloged many of the conspicuous and common taxa. Regular meetings (i.e. about every second day) were held during the scoring process to go through any new species observed to ensure consistency of labeling, and for the labeling process itself. The coordinator randomly checked images during the scoring process and again at the end to check that scoring was consistent between and within scorers. Any systematic issues found were addressed at this stage by re-scoring points with errors.

4.4.4 Queensland

A total of two people labeled this data set, from which one was an expert and one trained non-expert research students. Initially, both the scorer and an expert examined 40 images simultaneously for consistency in labeling. During the entire labeling process any uncertain labels were flagged and verified by the expert. At the end of the process data were plotted and checked for outliers, with any outliers
checked by the expert and re-scored if necessary.

4.4.5 Geolocation Accuracy

The positional accuracy of the AUV is described in [23]. Based on the sensors (including GPS on surface for an initial position fix with ship borne ultra-short baseline (USBL) positioning underwater), the accuracy is approximately ±5 m. The authors also describe several techniques based on visual feature matching that permit this to be reduced when comparing multi-year surveys in the same locations.

4.5 Usage Notes

4.5.1 Automated Labeling Research

The primary motivation for creating and releasing this data set was to enable progress towards automated labeling solutions for benthic imagery. In terms of supervised machine learning, the labels can be considered an example of a hierarchical classification problem [15]. A particular set of algorithms and approaches can be applied to automatically recognizing content within a taxonomy such as CATAMI. Alternatively, the problem can be recast as a single binary classification task (such as presence or absence of a particular class), or mutually exclusive set of classes (such as algae, coral, sponges and “other”).

Further, the points can be aggregated at an image level, in order to train and predict on overall habitat composition of images. Unsupervised techniques could also be used on the images to automatically detect visually similar groups of images.

A comprehensive review of the benthic computer vision literature up to 2012 can be found in [43] (in which Table 2.1 contains references to work focusing on classi-
fying whole images, and Table 2.2 covers point level classification in images). Since then, other relevant work includes [80], [76], [12] and [14]. The latter two look at the Tasmania 2008 campaign from this data set, with [14] performing hierarchical classification using a subset of the CATAMI hierarchy.

### 4.5.1.1 Geographical and Temporal Variation

One aim of this data set is to permit researchers to examine the robustness of their techniques to temporal and geographic variation. In terms of temporal variation, the Abrolhos region of Western Australia has data for dives in a small geographical area, from 2011, 2012 and 2013. Similarly, there is data from repeated surveys in New South Wales and Tasmania. In terms of geographic variation, the data set includes locations from a wide diversity of locations around the Australian coast. Common species such as *Ecklonia radiata* (kelp) are known to vary greatly in appearance between tropical and temperate areas, making for a very challenging machine learning problem. Aside from the sheer number of labels, this variety is the most unique feature of the data set. Without this type of collaborative data set, machine learning and computer vision researchers are typically restricted to working with a single data set from a particular bioregion, and making untestable assumptions about their algorithms’ ability to generalize.

### 4.5.1.2 Image Processing

As the AUV platform and survey process has matured over the years, changes have been made both on board *Sirius*, and in the post-processing of the data. Xenon strobes, for example, were replaced with white LEDs. The processed images represent the best processing available at the time the campaign was conducted, and importantly the processing that was used by experts when labeling the images. The
raw images can also be made available by special request, for those researchers wishing to investigate their own image processing techniques. The post-processing steps are described in [54].

4.5.1.3 Testing Performance

In order to facilitate easy and fair comparison of different approaches, a hidden test set (described earlier) has been reserved. Researchers wishing to publish performance results on the test set are encouraged to contact the authors.

4.5.2 Marine Science

Aside from automated labeling research, this data set also has potential value for direct scientific insights. By making a wide variety of label data available in a standard format, studies involving a comparison across time or geographic location become much more feasible, such as the recent study of *Ecklonia radiata* distribution around Australia [65]. Rather than solely comparing conclusions drawn by various studies at a high level (as a meta-study), the entire set of raw data is available for analysis as a single group.

Other marine science studies using similar types of data (up to 50 expert labeled points per benthic image) include [24], [21], [35] and [75]. One highly influential work used average abundances over large areas to demonstrate a 27 year decline of coral cover on the Australian Great Barrier Reef [31].
4.6 Acknowledgments

4.6.1 Author Contributions

Michael Bewley was the primary coordinator and author of the manuscript, and collated the data sets.

Ariell Friedman developed the Squidle platform, and contributed to manuscript development.

Renata Ferrari coordinated and conducted labeling of the NSW data set, contributed to development of the CATAMI classification scheme, and contributed to manuscript development.

Nicole Hill supervised the scoring of the Tasmanian imagery, contributed to the development of the CATAMI classification scheme, and contributed to manuscript development.

Renae Hovey contributed to the field campaigns and supervised the scoring of the Western Australian imagery, and contributed to discussions of the CATAMI classification scheme.

Neville Barrett led Tasmanian field campaigns, programs collating benthic data from acquired imagery, and contributed to manuscript development.

Oscar Pizarro took part in AUV operations throughout Australia, supervised data collection and processing, and contributed to manuscript development.

Ezequiel Marzinelli contributed to the field campaigns and scoring of the New South Wales imagery.

Will Figueira co-lead the group which conducted scoring of the NSW imagery, developed processes for determining optimal sampling frequency, and contributed to manuscript development.

Lisa Meyer developed the scoring methods and labeled a large portion of the Tas-
manian data.
Russ Babcock led Queensland field campaigns, programs collating benthic data from acquired imagery, and contributed to manuscript development.
Lynda Bellchambers contributed to expert labeling of the Western Australian data set.
Maria Byrne contributed to the manuscript development.
Stefan B. Williams took part in AUV operations throughout Australia, supervised data collection and processing, and contributed to manuscript development.

4.6.2 Other Acknowledgments

NH, NB and LM undertook this work as part of the Marine Biodiversity Hub, a collaborative partnership supported through funding from the Australian Government’s National Environmental Research Program (NERP). NERP Marine Biodiversity Hub partners include the Institute for Marine and Antarctic Studies, University of Tasmania; CSIRO, Geoscience Australia, Australian Institute of Marine Science, Museum Victoria, Charles Darwin University and the University of Western Australia. The Hub also facilitated the development of the CATAMI classification scheme.

The Marine biodiversity Hub for funding the scoring of imagery within its program and facilitating development of CATAMI.

The ANDS/ CATAMI project and researchers involved in developing CATAMI classification scheme.
Chapter 4  BENTHOZ-2015: An Australian benthic data set

The NSW Department of Primary Industries and the Great Barrier Reef Foundation for their support.

This is contribution 166 to the Sydney Institute of Marine Science.

4.7 Competing financial interests

The authors declare no competing financial interests.

4.8 Data Citations

Figure 4.2: CATAMI Hierarchy diagram. Numbers in brackets show the number of points in the data set that have been labeled as a given class, or one of its descendants. CATAMI Hierarchy has been extended to lower (species) level where appropriate data was available. Best viewed electronically.
Figure 4.3: Example of an expert labeled image in *Squidle*
Figure 4.4: Frequencies of most popular class labels appearing in each region
Figure 4.5: Distribution of annotated images over time

Figure 4.6: Distribution of annotated images over depth
Chapter 3 is based on a conference paper introducing the hierarchical classification problem in the context of automated analysis of benthic imagery. Here, those ideas are explored in greater depth, using the BENTHOZ-2015 data set described in Chapter 4 and [13]. Contributions are made to the general hierarchical classification literature by recognising that one of the most common solutions to the problem can be modelled as a Bayesian Network. We introduce Bayesian Network theory to explain assumptions and drive choices in a space dominated by heuristic solutions and empirical evaluation. In this chapter, we:

- Conduct a more extensive review of the hierarchical classification literature;
- Recognise the similarity between the local classifier per node approach with multiplication as a form of Bayesian Network;
- Apply Bayesian Network theory to define necessary assumptions and decisions (local classifier calibration, the siblings training set selection, etc.);
- Compare hierarchical performance, and ability to provide unbiased estimates of class abundance.
- Define a novel process of re-calibrating an inclusive trained set of classifiers.
using a sibling data set, in order to utilise the additional negative labels, without sacrificing the conditional probability assumption.

Abstract: Hierarchical classification problems constitute a subset of supervised learning, in which the output of a classification algorithm is structured as a taxonomy (directed graph), rather than a flat list of mutually exclusive alternatives. A wide range of largely heuristic approaches has been used in the literature. We apply a principled approach (using Bayesian Networks) to the commonly used local classifier per node framework, in which a single classifier is trained to recognise each class in the hierarchy. The use of Bayesian networks specifies certain aspects of the solution that are typically defined based on empirical performance, with little theoretical justification. Importantly, the conditional probability assumptions of the network highlight the need for proper calibration of local classifier probabilities, and suggest a novel approach of re-calibrating generically trained classifiers on a data set that satisfies the conditional probability assumption required by a Bayesian Network approach. Calibration performance is measured at a local classifier level, using the Calibration Score from the two-component decomposed Brier score. Performance of the system of classifiers as a whole is measured using the modified hierarchical f1-score, a modified hierarchical Brier score (loss function), and the accuracy with which class abundances are predicted. Performance is compared to other commonly used local classifier per node solutions on BENTHOZ-2015, a large hierarchical data set of scientifically annotated benthic imagery from an autonomous underwater vehicle, which finds significant practical advantages for the Bayesian Network approach.
5.1 Introduction

Hierarchical classification is a sub-type of the more general classification (or supervised learning) problem, in which a categorical label needs to be predicted from example data. In the simplest case of classification, the output is a single binary category, and the task is to predict whether or not a given example belongs to that category. Adding more complexity, the multi-class classification problem defines a list of categories (e.g. colours) which are mutually exclusive; the task is then to assign an example to one of these alternatives. This is also called “flat” classification, to distinguish it from the hierarchical case. In hierarchical classification, the desired output is a tree or directed acyclic graph (DAG), in which a node of the hierarchy represents a category, and the ancestors of that node represent more general categories. As such, the task is to predict the correct path from root node (with no ancestors) to some level of specificity in the hierarchy. This class of problems has been reviewed multiple times (most completely in [81]); we provide further recent background in Section 5.2.

On reviewing the literature, it is immediately clear that there are a wide range of approaches to solving the hierarchical classification problem. The two main groups include what is referred to as the “big bang” approach, where specific algorithms that solve the problem as a whole (such as Structured SVMs), and the “local classifier per node” approach, where an individual classifier is trained to recognise each class in the hierarchy. While the array of choices to be made in constructing a solution is clearly enumerated in [81], the literature tends to resort to empirical performance comparison to evaluate the various options.

In this study, we consider the local classifier per node approach as an example of a Bayesian Network\(^1\). The assumptions necessary to do this correctly have implica-

\(^1\)A Bayesian Network is a graphical model representing a joint probability distribution with
tions for the choices made during training a collection of local classifiers, such that selection of training and prediction strategies can be made on a principled basis, rather than an heuristic one. We then evaluate performance of the approach on a large hierarchical data set, consisting of benthic images captured by an Autonomous Underwater Vehicle, and labelled by marine scientists according to a scientific taxonomy.

The core contributions of the work are:

1. *A principled approach to hierarchical classification*: Recognising that a local classifier per node approach that restricts the negative training set to siblings of the positive class, and multiplies probabilities down the tree equates to a Bayesian Network in which the classifier outputs are assumed to represent the conditional probability of class membership of a given node.

2. *Calibration of classifier probabilities*: The conditional probability assumption implies that the classifier probabilistic output of the classifiers must accurately represent the likelihood of class membership, given an example is a member of the parent class. This concept of calibration is discussed seldom (if at all) in the hierarchical classification literature, but is crucial in order to meet the assumptions.

3. *Empirical validation*: We demonstrate on a large, real-world hierarchical data set that the principled approach performs at least as well as other commonly used heuristic ones, and that calibration has a measurable impact on performance.

Section 5.2 performs a brief review of the literature, building on work since [81]. Section 5.3 introduces Bayesian Networks, and works through their application to the hierarchical classification using the local classifier per node approach. Section 5.4
5.2 Background

5.2.1 Classification Problem Types

Table 5.1 defines terminology for 5 different types of classification problem. Readers will be familiar with binary classification, which involves either a choice between two classes (e.g. “day or night”), or presence detection of a single class (“True or False”). Flatt multi-class classification is the natural extension to greater than two classes, where each instance must be given one of a set of labels (e.g. “red, blue or green”). These two types of classification make up the bulk of the supervised classification literature. Further, the vast majority of classification algorithms are aimed at solving either binary or flat classification problems.

More recently, interest has grown in problems that do not conform to these definitions. A clear example of Multi-Label Classification is in photo tagging. A set of potential tags is defined (e.g. “holidays, nature, pets, urban”). Extending beyond flat classification, multi-label classification permits any number of classes to be active; a photo may represent a holiday in nature, or pet in an urban environment. Note that while certain tags may be related (nature photos may be more likely to be taken while on holidays), there is no fixed set of rules about what combination of tags are permitted to be activated on a given example.

Hierarchical Classification is a combination of flat and multi-label classification. The output classes are organised into a pre-defined hierarchy (either a tree or DAG relationship), where the directed links denote an “is-a” style membership in which the
parent is a superclass, and the child is a more specific sub-class of the parent. Only a single output class may be explicitly activated for an instance, with the chain of all ancestor classes implicitly activated. In a sense, therefore, hierarchical classification is a subset of multi-label classification, where multiple labels are permitted, as long as they satisfy the constraints of the class hierarchy.

Yet another modification is to permit multiple classes to be explicitly activated, and require all of their ancestor classes to be implicitly activated; this is known as multi-label hierarchical classification.

### 5.2.2 Attributes of a Hierarchical Classification Model

A recent study [81] conducted a review of a broad body of work in the hierarchical classification field, and defined a framework and terminology for discussing solutions to the problem. In summary, they define four attributes that describe the capabilities and approach of a hierarchical classification model.

#### 5.2.2.1 Single vs Multiple Path Prediction

Single Path Prediction (SPP) is used to describe a hierarchical classifier that only permits a single chain of nodes to be activated, from the most specific, back through
the ancestors to the root node. Multiple Path Prediction (MPP) allows more than one class to be explicitly activated for the one instance (each of which will have its own ancestor path implicitly activated). In terminology used elsewhere in the literature, SPP and MPP are described as *hierarchical* and *multi-label hierarchical* classification respectively.

### 5.2.2.2 Mandatory vs Non-Mandatory Leaf Node Prediction

Mandatory leaf node prediction (MLNP) implies that prediction must always progress to leaf node level. Non-Mandatory leaf node prediction (NMLNP) permits the prediction to end at some partial depth of the tree, without necessitating a prediction to leaf node. In the example representing a child’s toys, MLNP may require a label as specific as “soccer ball”, whereas NMLNP may stop prediction at “balls”, or even “outdoor toys”. In both cases, it is generally assumed that the true labels of the data have been supplied at leaf node level; in a real-world data set such as BENTHOZ-2015, this is not necessarily the case.

### 5.2.2.3 Tree vs DAG Structure

A tree structure is defined as having a single root node, which has child nodes. Any node may have any number of children, but may only have a single parent. By contrast, a DAG permits nodes to have multiple parents. In a DAG, cycles are still not permitted (i.e. starting at any given node and following directional relationships can never permit you to return to that node). Tree structures are the most commonly studied, and are a subset of the DAG problem.
5.2.3 Types of Hierarchical Classification Frameworks

Whereas the previous section describes subtly different definitions of the problem, this section describes three broad categories of frameworks used in solving hierarchical classification problems. This work focuses on the local classifier approach, however a discussion of all three is included for completeness.

5.2.3.1 Flat Classifiers

The simplest approach to hierarchical classification is to reduce the problem to one of flat classification. By selecting a set of classes with mutually exclusive definitions (no parent/child relationships between them), any multi-class classifier (or indeed binary classifier using an approach such as one-vs-rest voting) can be used to solve the problem. While this simplifies matters, it imposes significant limitations. Firstly, any time a different set of classes is of interest, the classifiers must be retrained. Consider a classifier that identifies species within a scientific taxonomy; a researcher interested in insects requires a different classifier to one interested in bird species. Performing hierarchical classification on a carefully designed semantic hierarchy can allow a user to “roll up” the classes that are not of interest, and explore the areas of the tree that are relevant to a particular usage. There is still some risk of retraining in hierarchical classification, as solutions are only flexible within the same semantic framework (e.g. a hierarchy based on visual appearance of organisms compared to genetic similarity).

The second disadvantage to simplifying the problem to the flat case is that any semantic information embedded in the hierarchy is lost, and unable to be leveraged by the classification algorithm to improve performance. [15] discusses the difference between semantic and feature space similarity (in their case visual similarity for
images), and the knowledge embedded in the semantic hierarchy that may improve classifier performance.

### 5.2.3.2 Global Classifiers

If the class hierarchy is to be preserved, it is necessary to create an algorithm that provides a hierarchy as an output, rather than a binary or categorical value. A *global classifier* (also described in the literature as the “big bang” approach) seeks to do this via the creation of a specialised classification algorithm. Structured Support Vector Machines [90] are an example of this approach. In the computer vision literature, this is often referred to as *structured output prediction*. The output of the classifier is a binary vector of classes, which in the most general sense can be used to solve any of the hierarchical classification problem types in Table 5.1. The misclassification cost function to be minimised is typically the number of different bits in the output vector between the true and predicted output (also known as the Manhattan distance, or L1-norm of the difference vector). In hierarchical classification terms, this is identical to the path length in the hierarchy between the true and predicted class. This neatly captures the sense that distant objects in the hierarchy (such as poodles and parrots) should be penalised more harshly than mistakes nearby in the tree (such as poodles and spaniels). Under the structured output classification interpretation, the class hierarchy becomes simply a method of defining the desired misclassification cost between any two classes. In this sense, structured output classification (when constrained to predict according to a hierarchy) can be generalised to *cost sensitive classification*, in which a flat multi-class problem with a single categorical output has a different cost of misclassification between each pair of classes.

This perspective highlights the fact that there is no theoretical reason for using the path length (or number of differing bits) as a cost function. Many papers cite the
intuition that misclassification penalty between something like a chimpanzee and a gorilla should be lower than that between a chimpanzee and a slug (e.g. [15]), however this does not mean that the cost must be precisely the path length between the two classes. Hierarchical classification is somewhat arbitrary in many cases. In a scientific taxonomy, the nodes can be arranged by morphology, or by genus and species, or by visual similarity, or even geographic presence. In essence, the hierarchy is an intuitive way for the human expert to define and visualise the misclassification cost matrix between a large set of classes. Taking this more generally with structured output prediction, it would be entirely possible for an expert to assign some level of similarity between two vectors of classes using another metric, or even in a completely manual way (such as using survey results on perceived similarity).

Using other distance penalties is also possible to achieve various effects, e.g. using the Euclidean Distance (the square root of the sum of squares, which equates to the path length between true and predicted nodes) to de-emphasise the penalty induced by very long path lengths.

5.2.3.3 Local Classifiers

A global classifier uses a specialised algorithm to predict either a binary vector as output, or a categorical output with specified misclassification costs. By contrast, a local classifier breaks the problem down into a set of binary classification problems, and then aggregates the output. While other variations exist, the most popular is to train a single binary classifier to recognise each class in the hierarchy - usually referred to as the local classifier per node approach. In the example of a scientific taxonomy, one classifier might be trained to recognise biological vs physical objects, another might recognise plants and animals, and another might recognise a particular species of animal.
A clear disadvantage in breaking the problem down in this way is that it prevents the local training algorithm from optimising directly for a misclassification cost matrix (e.g. path length difference), as each binary classifier is trained independently. The outputs of the local classifiers must then be combined in some way such that the hierarchical classification definition (of only activating unbroken chains of classifiers from predicted to ancestor nodes) is adhered to. The advantages of the approach are described in detail in Section 5.2.4 below.

### 5.2.4 A closer look at Local Classifier Per Node frameworks

This study deals largely with an exploration of the *local classifier per node* framework. We explore the large number of possible ways to train the local classifiers, then combine their outputs for prediction such that they operate effectively as a single aggregated classifier. As previously noted, attention in the literature has been focussed largely on exploring the method with the best empirical results, rather than seeking to ground the decisions on some theoretical basis.

One clear benefit of the *local classifier per node* approach is the ability to re-use existing classification algorithms as the node classifiers. SVMs, logistic regression, random forests, or any other binary classification algorithm can be used to solve the local classification problems, whereas a global approach requires a specifically tailored classification algorithm that takes class structure into account.

As well as flexibility in classification algorithms, the definition of predictive features (attributes), can be customised and optimised for each sub-problem. There is no constraint requiring the same set of features and binary classification algorithm at each node. Further, it is even possible to move away from a purely supervised learning approach, to including expert systems, or any other valid means of predicting the output class. One could easily imagine that while a supervised learning ap-
proach may provide excellent discriminative power between plants and sand, it may be almost impossible to perform classification between two visually similar plant species without some level of expert knowledge applied. In [73], an approach is to plant species classification is discussed, that uses expert derived features based on leaf dimensions and vein structures within the leaf. These features are entirely irrelevant for distinguishing between plants and sand (and indeed cannot meaningfully be computed for sand), and are unlikely to be learned by feature learning techniques based on the number of training instances available for a rare class. It may be intellectually unsatisfying for the machine learning practitioner to resort to such expert systems; however hierarchical classification often pushes the boundaries of training with small amounts of data, as the amount of training data reduces approximately exponentially as the tree depth is increased further for finer grained classification. Appropriate use of expert knowledge may permit classification at greater depths than is possible using automated algorithms alone.

The local classifier per node approach also permits the use of training data that has not been labelled to leaf-node depth. A global classifier would need to build in the concept of unknown/unspecified bits in the output vector (which is possible, but yet another constraint requiring a particular specialty algorithm). In the example of marine science, experts tend to label to a depth in the hierarchy that is commensurate with their area of expertise, scientific aims, resources or quality of the data. This results in data sets where the labeller may, for example, carefully label algae to the species level, and provide only approximate labels to the general groups of coral, fish or sponges. Another labeller may have a particular interest in morphology of sponges, and only label coral and algae approximately. Both of these use cases can be applied to the same class hierarchy, as long as the classifier is able to be trained using data labelled to partial tree depth. Data sets labelled to full depth
tend to be more common, as they are easier to work with, and the simpler choice when constructing a synthetic data set. There is also a natural propensity to define a class hierarchy to such a depth that only few, rare instances of the leaf classes are identified. BENTHOZ-2015, the marine imagery data set used to evaluate this study, includes a large amount of partially labelled examples due to its origins as a composite data set from working marine scientists. While hierarchical classification is typically simplified to include only leaf-node labelled data, partial depth labelling is an important use case in problems where it is impractical to provide all labels at the most specific level.

5.2.4.1 Training Strategies for Local Classifiers

In defining the positive and negative training examples for each node, Silla and Freitas define several approaches [81]. The two most prominent approaches described are the inclusive and the sibling policies.

- **The inclusive training policy:** This policy includes as positive examples the entire subtree of the training node (i.e. every node with the training node as an ancestor). The nodes in the rest of the tree are used as negative examples, with the exception of direct ancestors of the training node. The direct ancestors cannot be used as either negative or positive examples as they contain instances of both; e.g. if training a classifier for node *ALGAE* in Figure 5.2, then node *ALGAE* is used as positive examples and *SPONGES* and *PHYSICAL* as negative examples.

- **The siblings training policy:** This alternative policy uses the same positive training examples, however, the negative examples are restricted to siblings of the training node (and not siblings of the ancestor nodes). Using the example from before, the negative training samples will come from *SPONGES* only.
The expected performance difference between these two policies is not obvious, with no consistent winner found in [40]. On one hand, the inclusive policy ensures that each node is as informed with as much data as possible, and should be able to deal better with classifying instances that belong elsewhere in the tree (as it has seen examples of all the data). On the other hand, the siblings policy solves a much more specific problem (distinguishing a node’s class from its siblings), and may give better discriminative performance between these classes. Such nodes will, however, be less informed about instances that belong elsewhere in the tree. An inherent advantage of the siblings approach is that it scales far better with tree size, as nodes deep in the tree use significantly less data during training. Related to this, the classifiers in the inclusive approach will have an increasing level of class imbalance with depth. [14] found that in training a large number of models, siblings and inclusive performed similarly, with superior performance for inclusive in the very best models.

5.2.4.2 Prediction Strategies (aggregating local classifier outputs)

Given a set of locally trained classifiers, it is necessary to create some strategy for aggregating their outputs. Complete consistency is required in the hierarchical labels, such that a single, unbroken chain of classifiers predicts positive results from the root to the deepest node. This consistency is broken if the local classifier outputs are taken as-is (resulting in nonsensical claims such as an example being classified as simultaneously a “soccer ball”, a “toy car”, and “not sporting equipment”).

A wide range of aggregation approaches have been used in the literature, which are largely heuristic in nature, with empirical justification provided for the choice.

- **Greedy Walk**: Starting with an instance at the root node, the child predicting greatest probability of membership is chosen. The node with the greatest

---

3Described as Maximum Probability Switching in Chapter 3 and [14]
5.2 Background

probability of membership among the children of that selected child is chosen, and so on until a leaf node is reached. Alternatively, a stopping criterion can be applied when the highest child probability for an example drops below a threshold. This is in effect a greedy algorithm, making locally optimal decisions at each point in the tree. The weakness is that errors higher up in the tree have an irreversible impact on the correctness of the result; an incorrect decision at the top level of the tree means that even if the rest of the classifiers on the correct path are highly accurate, their results will be completely ignored.

- **Probability Aggregation**: This weakness of the greedy walk often prompts a search for alternative methods that are more robust to erroneous choices at the top of the tree. [50] compare three methods of probability aggregation between the root and most specific prediction node: Multiplication, summation, and ranking of probabilities. Each node has a binary classifier trained, in what amounts to the inclusive method described above. MLNP is also used in their formulation. They conclude that performance in all three aggregation approaches is similar to the performance of the greedy walk approach. However, the performance was measured only in terms of precision (both standard and hierarchical), with no consideration of recall. This is problematic, given the precision can usually be increased at the expense of recall, by moving the decision boundary in the more conservative direction.

- **Normalising for path length**: The weakness of the multiplication approach is that it inevitably penalises prediction deeper in the tree, as each subsequent prediction lies in the range $[0, 1]$. As the net probability of a child node is always less than that of its parent, the MLNP restriction cannot be lifted for multiplication (it would result in no predictions deeper than the first layer), and is biased against predictions in parts of the tree with greater leaf node
depth. This is addressed in [15] by normalising for path length. Experimenting with generalised $p$-means of ancestor path probabilities, they find the geometric mean of probabilities to have the best empirical performance on their data set.

### 5.2.5 Hierarchical Performance Metrics

Having trained a classifier that is capable of making predictions on new examples, it is necessary to have some method of measuring performance; ideally a single number to evaluate the performance on an entire tree, such that selection between a large number of models can be automated. The reason that flat performance metrics are unsatisfactory is described succinctly in [15]: “...humans tend to perceive some confusions like cat versus fridge to be more unnatural than others like cat versus dog which can be reflected by a taxonomy”. A performance metric which cannot penalise mistakes in the hierarchy appropriately is unlikely to give meaningful results, and (when optimised during parameter tuning) to result in a poorly performing classifier.

For global classifiers, the performance metric should be defined by using the same misclassification cost applied during training. This is an advantage when compared to local classifiers which are not trained to optimise a misclassification cost in the same way as the global classifiers.

A global classifier that produces a vector of classes is is typically assessed using a loss function that equates to the number of incorrect classes in the output vector. In the more specific context of hierarchical classification, this equates to the path length in the class hierarchy between the true and predicted node. Figure 5.1 depicts this in terms commonly used in flat multi-class classification; true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). It is simply necessary to consider that each testing instance may have multiple counts assigned
to it, according to the true and predicted locations in the output hierarchy. Each node in the chain of true class nodes is compared to the chain of predicted nodes. 5.1a and 5.1b illustrate two scenarios. In the former case, the prediction down the entire tree is accurate. In the latter case the last prediction is incorrect. This results in a FN detection as the correct node was missed as well as a FP detection for an incorrect node being detected.

When employing the greedy walk, it is possible for the prediction to stop earlier up in the tree than the ground truth, when a probability threshold is used. In such an instance, the missed nodes are calculated as FNs. This is illustrated in 5.1c.

Given a hierarchical version of each of these, it is then trivial to compute other standard performance metrics over the hierarchy such as the hierarchical precision \( p = \frac{TP}{TP + FP} \), hierarchical recall \( r = \frac{FN}{TP + FN} \), hierarchical f1-score \([56]\) \( F_1 = 2 \cdot \frac{p \cdot r}{p + r} \), or simply the mean loss across all examples and classes.

A modification to this approach was introduced in [14], for cases of partial depth labelling, where the true class may (or may not) be a descendant of the expert derived label. It would be unfair to penalise in these ambiguous cases, as they are not necessarily errors. No penalty (FP or FN) or reward (TP or TN) is applied for incorrect labels that descend from the true label. This modification can be easily included in any of the hierarchical classification performance metrics described in this paper.

### 5.3 The Bayesian Network Interpretation

Those familiar with probabilistic graphical models may note the strong similarity between Bayesian Networks and the multiplicative prediction of the local classifier per node frameworks discussed in Section 5.1. As discussed in Section 5.3.1, this link
Figure 5.1: Hierarchical performance metrics. The connected circles represent part of a class hierarchy. Grey circles are ground truth, white circles are prediction. 5.1a shows the case of the classifier network correctly predicting a node of depth 3 in the hierarchy. 5.1b represents the first two levels being predicted correctly, whereas the classifier network incorrectly chooses a sibling of the true class at depth 3.

does not yet appear to have been made explicitly in the literature. This section explores that relationship in detail, and the associated constraints and implications of considering *local classifier per node* hierarchical classifiers as Bayesian Networks.

5.3.1 Related Work

The work in [59] introduces an approach using Bayesian Networks on a hierarchical document classification data set (Reuters-22173), to select successively smaller subsets of features for node level classifiers. At prediction time, however, the authors use a simple greedy walk to classify documents, noting that errors made at the first layer of the hierarchy are “unrecoverable”. This is built on by [36], by an application to two layer hierarchical classification of web content, using both a “multiplicative” and “boolean” approach to prediction, without mention of Bayesian Networks. Further work on this line is performed in [32] (modelling xml document structure with
Bayesian networks), and [30] (a Bayesian Network used is for hierarchical document classification based on a thesaurus). This group of approaches are essentially an extension of the Naive Bayes classifier to more complex networks; they do not make use of local probabilistic classifiers in the same sense as a local classifier per node solution; rather they estimate the conditional probability distributions based on frequencies of words and terms directly.

A framework is introduced in [8], which is utilised extensively in Chapter 6; that of a Bayesian network with separate nodes for the true class membership, and for classifiers trained to predict those classes. This is the only prior work of which we are aware which explicitly uses Bayesian Networks for prediction. The structure, however, does not match the simple multiplication of probabilities used in most hierarchical classification studies (such as [92], which makes use of the multiplicative approach for prediction using a network of decision trees in the field of functional genomics). Brief mention is made of the concept of representing conditional probabilities via classifier outputs, by selecting the siblings training set, and using the product rule (i.e. multiplication) and thresholding for prediction. Follow up work in [57] replaces decision trees as the node classifiers with ensembles, and continues the use of the product rule for prediction. In a review of the literature, [57] notes that the related field of multi-task learning (in which an autonomous agent is seeking to learn a set of related tasks) makes heavy use of Bayesian theory. The only work cited in the extensive reviews of [81] and [57] that applies Bayesian theory to hierarchical classification is [8].

As mentioned earlier, [15] uses generalised p-means for prediction, and explicitly mentions that a probabilistic interpretation of the approach is not possible, given they opt for the inclusive rather than the siblings restriction of training examples. While [25] appears relevant, the underlying approach make use of successively
thresholding classifier during prediction, so is not using Bayes theorem to perform classification in the same way as we describe here.

### 5.3.2 Bayesian Networks

A Bayesian network is a graphical representation of a joint probability distribution over a set of random variables. Directed links indicate conditional dependence between two variables (in the case of binary variables, the probability of the child conditioned on the value of the parent). Consider the graph in Figure 5.2 as a Bayesian Network over the binary variables $Al = Algae$, $Ph = Physical$, etc. The joint probability distribution would be described by:

$$P(Ro, Bi, Ph, Al, Sp) = P(Ro)P(Bi|Ro)P(Ph|Ro)P(Al|Bi)P(Sp|Bi)$$

This factorisation relies on the simplifying assumption that each variable is independent of all other variables, if the state of its parent is known. At each node, a conditional probability table is defined, as shown in Figure 5.2.

The conditional probability tables in the example represent information such as the
5.3 The Bayesian Network Interpretation

following:

- If the object is not biological, there is 0% chance that it is algae or sponge (by definition)
- If an object is biological, there is a 40% chance it is also algae
- There is a 62.5% chance of an object being biological, and a 12.5% chance of being physical

5.3.2.1 Inference

Inference is the process of calculating the overall probability distribution of a variable, given other known variables. In a tree structured Bayesian Network of binary variables, the final (posterior) probability for a node can be calculated trivially by multiplying ancestor probabilities.

\[ P(Al) = P(Al|Bi)P(Bi) \]

In this example, with no further information, the probability of an object being algae is \( P(Al) = 0.625 \times 0.4 = 0.25 \).

5.3.3 Local Classifier Per Node as a Bayesian Network

Under certain assumptions, the example in Figure 5.2 can be considered as an example of local classifier per node hierarchical classification. The binary variables represent whether or not an object belongs to that class (e.g., *Biota*). The chosen values in the first columns of the conditional probability tables enforce the requirement of all ancestors of the true node being activated, by assigning a probability of zero to the child when the parent is not activated (it is impossible by definition for a piece of algae to be non-biological).
When the parent is activated, it is necessary to assign a probability of the child also being activated. In a typical expert system, where a human has designed the network, the conditional probability tables will be constructed by either using the proportions of each class in the data set (the frequentist approach), or using the data to update a prior on each of the distributions (in the Bayesian approach). Either of these techniques provide a good baseline for estimating, based on the data set, how likely an example is to belong to any given node in the class hierarchy. Further, exact inference can easily be performed to update this information if some of the values are known.

Hierarchical classification, however, is about supervised learning. In a local classifier per node approach, we might obtain an estimate for the probability that an object belongs to a particular node, based on some set of features or attributes (in this example, perhaps image data is available for use in classification). The key step in applying a Bayesian Network is to assume that the probabilistic output of the classifier is a good estimate for the conditional probability of node membership, given the parent. Rather than using fixed values in the conditional probability tables, the values in the right hand columns of Figure 5.2 are set dynamically for each test instance, based on the local classifier outputs. Implicit Bayesian Network seems an appropriate term to use for such a model, and is used for the remainder of the thesis.

### 5.3.3.1 Implications for training and prediction choices

This Bayesian Network interpretation of hierarchical classification is bound by the following constraints:
Ancestor path activation: For a given node to be activated, all ancestors must be activated. This enforces the fundamental assumption of hierarchical classification, that ancestor nodes represent supertypes of their child classes (“is-a” relationships). This can be achieved, as in the example, by assigning 0% probability of child class membership, given it is known that an example is not a member of the parent class. This is an unusual approach for Bayesian Networks, as extreme probabilities are typically avoided (as they can prevent an outcome from occurring regardless of the amount of evidence being supplied to support it). This behaviour is desirable here, in order to strictly enforce the hierarchical definition.

Single or Multiple Path Prediction: SPP (where only one node can be explicitly activated) is not enforced by the framework. Knowing that an object is a sponge, means that it is also known that it is biological (due to ancestor path activation). This knowledge has the interesting side effect of actually increasing the probability of algae from 25% to 40%, rather than ruling it out as a possibility, which is somewhat counter-intuitive. SPP can be enforced after the fact, however, by applying methods such as greedy walk.

Non mandatory leaf node prediction: Conditional probabilities for sibling nodes (given the parent) do not need to sum to 100%, as the probabilities of the sibling nodes are conditionally independent given the parent node. This means that a parent class can represent a superset of the child classes that have been explicitly defined. In our example, an object that is neither sponge nor algae may still be biological. This allows the flexibility of not having a complete definition of the class hierarchy up front.
**Siblings rather than Inclusive training:** The *siblings* approach fits the assumptions more closely than the *inclusive* one; *siblings* classifier probabilities represent the conditional probability given the parent class [92, 57], whereas the *inclusive* training scheme does not [15]. Using the subtree of the given class as the positive training examples, and the sibling subtrees as negative, all training examples are at least members of the parent class. A probabilistic classifier then predicts the likelihood of membership of the class given the parent, implicitly satisfying the assumption. According to current practice, then, one must choose between using *siblings* to satisfy the conditional probability assumption and reduce training times, and using the *inclusive* approach to increase the amount of training data with potential performance benefits. In Section 5.3.4.1, a novel approach is introduced to use classifier calibration to permit the *inclusive* training policy to also represent the conditional probability.

**Prediction by multiplication:** In a tree structured Bayesian Network, which is simply a graphical representation of a joint probability distribution, multiplication of probabilities on the ancestor path is the only correct way to perform inference. Deeper nodes in the hierarchy will always have lower probability than their parents, due to the multiplication of probabilities in the range [0, 1]. Although this removes the ability to simply select the node with highest probability as the best estimate (this will always be one of the top level nodes), it permits a probabilistic interpretation of the output, without hard classification being required. If we have some level of confidence that an object is *Algae*, we should always be more confident that it is at least *Biota*, a more general class.
5.3.4 Practical Implications

After defining the set of assumptions necessary to treat local classifier per node, and exploring the constraints, there are a number of practical implications to consider regarding implementation and usage of such models.

5.3.4.1 Classifier Probability Calibration

A classification algorithm such as logistic regression is capable of providing an estimate in the range $[0, 1]$ that notionally represents the probability that a given example is positive (belongs to that class). The assumption of an implicit Bayesian Network is that this probability represents the likelihood that an example is a member of the class, given the parent. This raises the important question of probability calibration, which asks to what extent the predicted probabilities represent a fair estimate of the observed rate of positive instances (e.g. can one expect to observe one in five positive cases amongst a group of instances that have a mean probabilistic prediction of 20%). This can be quantified through the use of the two-component decomposed Brier score\cite{67, 17}, or a “reliability plot”\footnote{Also known as a “calibration plot”} (first columns in Figure 5.7), which is effectively a visual representation of the decomposed Brier score.

In this section, a metric to quantify calibration is described (based on the Brier score), as well as a method to correct it. While the correction is typically used to readjust a poorly calibrated classifier, a novel application is made to recalibrate classifiers using the inclusive approach on a siblings calibration set. In effect, this permits the local classifiers to be trained via either approach, and recalibrated on siblings data to meet the conditional probability assumption.
**Reliability Plots:** Whether a classifier’s probabilistic output is well calibrated can be determined from a reliability plot[91, 69]:

1. A calibration set is taken away from the data set prior to training, and is used both for assessing and correcting calibration.

2. The calibration set is divided (by convention) into decile groups of probabilistic predictions (e.g. all examples receiving estimates of 0-10% from a given classifier, will fall into the bottom group).

3. The mean predicted probability in each group is plotted against the actual fraction of observed positives in the group.

4. The $y = x$ line represents optimal calibration, where the mean probability in a group perfectly matches the fraction of positives observed within the group.

**Brier Score:** The Brier score is used to assess the accuracy of probabilistic predictions of a classifier. It incorporates both the discriminative power of a classifier, and how well calibrated the probabilities are. In its most common form, it can be computed by $B = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$, where $\hat{y}_i$ is the predicted probability for an example, $y_i$ is the true binary outcome for the example, and $N$ is the number of examples in the test set.

**Calibration Score:** This score can be decomposed as $B = C + R$, where $C = \frac{1}{N} \sum_{k=1}^{K} n_k (\hat{y}_k - \bar{y}_k)^2$ and $R = \frac{1}{N} \sum_{k=1}^{K} n_k (\bar{y}_k (1 - \bar{y}_k))$. $C$ is the calibration score, and relates to the calibration of the probabilities. $R$ is the refinement, which measures the overall discriminative power of the classifier, and is related to the ROC AUC. $K$ is the number of unique probabilistic predictions, and $\hat{y}_k$ is the observed frequency of the positive class within the set of examples having prediction $k$. In the case of a classifier producing real valued probabilities, an approximation can be made by
creating a finite number of nearby probability groups, and using the mean prediction within each group as $\hat{y}_k$. In this sense, the calibration score becomes the sum of squared differences in the reliability plot (as in e.g. the left hand columns of Figure 5.7), weighted by the number of examples in each group. Concretely, the $\hat{y}_k$ are the bin means of the predicted probabilities (plotted on the x axis), and $\bar{y}_k$ are the observed frequencies within the bins (plotted on the y axis). It is worth noting that the Brier score can be computed both for hard classification (in which case it is identical to the number of errors), and soft classification (in which case it is the sum of squared differences between the probabilistic prediction and true value).

These tests account only for calibration over the data set as a whole. Variation in performance within different sub-groups of the data set (in the species example, perhaps examples from different geographic regions) will not be accounted for. [91] notes that at least in ecological modelling, calibration is rarely tested for, let alone corrected, despite the enormous implications for class abundance accuracy.

**Correcting calibration:** Taking the frequentist approach, the probability of class membership given parent membership (and no other information) is simply $\frac{N_{\text{child}}}{N_{\text{parent}}}$, where $N_x$ denotes the number of instances belonging to the subtree of classes under node $x$. Over the whole dataset, then, the average probability over all instances should equal the frequentist conditional probability.

When this is not the case, the probabilistic output of the classifiers can be adjusted to compensate, making the conditional probability assumption more reasonable. This process of adjusting the probabilistic output of a classifier is usually referred to as “calibration” [69].

In [91], the authors discuss the impact of poor calibration of binary classifiers on species abundance estimates. In ecology, it is common to take randomly distributed
samples, and extrapolate to investigate abundance in a wider geographical area. If a model is used to predict the probability of species presence at a point, the weighted sum of probabilistic estimates should result in a good estimate of the total prevalence of that species in the wider area. The authors comment that it is rare to see evidence of calibration testing, and ascribe this to a reliance on e.g. the AUC ROC score (which measures discriminative power of the classifier), which does not necessarily imply good calibration. This issue is important not just for prevalence estimates in ecology, but also for hierarchical classification, given our assumption of classifier output representing conditional probability. If multiplication is used to perform inference, and then the MAP estimate is used (to select the node with highest probability prediction), it is crucial that the probabilistic output of every classifier is directly comparable with the others. Even a single path in the hierarchy that provides inappropriately high probability estimates will dominate all other classes.

The two most commonly used calibration methods are Platt Scaling (fitting a sigmoid to the probabilistic output of the classifier), and Isotonic Regression (a more flexible fitting approach that specifies only that the function must be monotonically increasing). [69] performs a detailed assessment on a range of classifiers using these two approaches, and finds that Isotonic Regression (using a stepwise-constant algorithm) outperforms Platt Scaling when a larger calibration set is used, but tends to overfit with less data, due to its ability to fit any monotonically increasing curve close to exactly.

5.3.4.2 Class Imbalance

Class imbalance is defined as the ratio of observed classes in the data set. It is frequently avoided by assuming that the data set used for training and evaluation has at least approximately similar observation rates of each class.
In hierarchical classification, class imbalance is inevitable. With perhaps hundreds of classes, even equal numbers of examples from every class results in low observed per class frequencies. With the siblings training set suggested by the Bayesian Network assumption (training local classifiers on sibling data only), this problem is substantially reduced for individual local classifiers, unless the tree includes classes with many children. In a real world data set, however, it is likely that some of the classes will be very common, while others remain exceptionally rare.

Common approaches to training classifiers on unbalanced data include over or under sampling examples (to enforce an equal number of positive and negative training instances), or preferably weighting examples such that the misclassification cost is proportionally higher for the smaller class. This weighting approach is possible for many classification algorithms (logistic regression, support vector machines, random forests), and can result in substantially improved discriminative performance (as measured by ROC AUC).

When classifier probabilities are used, it is clear that under the Bayesian Network assumption, the probabilities must be calibrated to the expected observed conditional frequencies of each class given the parent. For other approaches (e.g. greedy walk) that require binary classification, it is necessary to choose an appropriate threshold for the classifier, based on optimisation for one of a large number of performance metrics such as accuracy, f1-score, balancing sensitivity and specificity, or simply using a threshold of 50% probability.

**Precision and performance limitations on unbalanced data:** The precision of a classifier is defined as the fraction of positively classified instances that are correctly labelled, or $TP/(TP + FP)$. The vertical axis on a reliability plot is the observed relative frequency of positive instances within each predicted probability group (usu-
Table 5.2: Unbalanced class confusion matrix. \( n \) is the number of positive instances, \( s \) is the balanced sensitivity and specificity, and \( b \) is the class imbalance factor.

<table>
<thead>
<tr>
<th></th>
<th>Predicted Negative</th>
<th>Predicted Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>( bns )</td>
<td>( bn(1 - s) )</td>
</tr>
<tr>
<td>Positive</td>
<td>( n(1 - s) )</td>
<td>( ns )</td>
</tr>
</tbody>
</table>

ally decile). Effectively, the reliability plot is showing the precision for each group, assuming that the group is in turn considered as the set of positive predictions.

On highly unbalanced problems, trading off between precision and recall (usually via the f1-score) is typically much more challenging than between sensitivity and specificity. Consider a classifier with the threshold chosen to balance equal sensitivity and specificity (at value \( s \)). Let the number of positive examples be \( n \), and the imbalance factor \( b \) such that there are \( bn \) negative examples. This results in:

\[
\text{sensitivity} = \text{specificity} = \text{recall} = s, \quad \text{precision} = \frac{s}{s + b(1 - s)}, \quad \text{and} \quad f1 = \frac{2s}{2s + 1 - s + b - bs}
\]

The ROC AUC score is closely related to the sensitivity and specificity. The confusion matrix for this scenario is shown in Table 5.2. Precision, and therefore f1-score depend on the level of class imbalance \( b \), whereas sensitivity, specificity and the ROC AUC score do not. For equal positive/negative class balance \((b = 1)\), precision and recall are equal \( (s) \). As shown in Figure 5.3, increasing the imbalance in the data drastically reduces the ability to achieve high precision, and a classifier with \( s \approx 90\% \) and \( b \approx 100 \) will have a precision of only \( 8\% \). For a well calibrated classifier with fixed \( s \), this means that the distribution of probabilities will be skewed heavily towards zero, and the calibration will therefore be less accurate for the more rare higher probabilities.
5.3 The Bayesian Network Interpretation

![Precision with Unbalanced Data](image)

**Figure 5.3:** Precision with unbalanced data. Colour and contours represent precision at a given value for imbalance ($b$) and sensitivity/specificity ($s$).

5.3.4.3 DAG and Tree structured Classes

While many hierarchical classification systems are only designed to work with tree structured classes, a Bayesian Network immediately permits DAG structures to be used, with no modifications or heuristic concessions. Permitting nodes to have multiple parents allows much more complex semantic hierarchies to be used.

Under these assumptions, performing *siblings* training with multiplication of probabilities is the most theoretically sound approach to training and prediction of *local classifier per node* hierarchical classifiers. There is no basis other than heuristic suggestion and empirical performance evaluation for greedy walk, addition, or geometric mean to be used for prediction.

The question remains as to whether this interpretation has practical value, or is merely theoretically pleasing. We assess this by examining both useful properties of Bayesian networks (below), and empirical performance (in Section 5.4 and Section 5.5).
5.3.4.4 Properties

When considered as a Bayesian Network, the following properties result from the theoretical framework:

- Aggregated classifier outputs can be directly interpreted as the probability of a given example being a member of that class. This is not possible under any of the other heuristic frameworks.

- Prediction of more general DAG structured class hierarchies can be performed, with the same theoretically sound approach.

- If some of the true states of the nodes are known from extra information (i.e. that an example is “definitely not” one class, or is “at least” as specific as another), the probabilities can be substituted, and prediction can still be performed.

- If the classifier output is unavailable for a particular node, the conditional probability for that node can be estimated from the data (i.e. \( P(n_{\text{child}}|n_{\text{parent}} = 1) \) is the fraction of cases in the training set that belonged to \( n_{\text{parent}} \), that also belonged to \( n_{\text{child}} \)). In cases for which different expert derived features are used for different nodes, it is possible that the feature cannot be computed for a given node. Similarly, in the underwater robotics application in Section 5.4, a sensor may become briefly unable to compute results, resulting in missing values in the classifier outputs.

- If no strict hierarchy has been provided, it is possible to learn the structure of the class hierarchy from the data, even if not every single node membership status is known for a given example. Learning Bayesian Network structure is a well established process, typically involving statistical tests of conditional independence between variables [78].
• In problems where abundance of the class is important, properly calibrated classifier outputs can be used to provide an unbiased estimate of class frequency, which is an important problem in problems such as abundance of species in ecology. The average probability score at a node (using inference via ancestor multiplication) represents an unbiased estimate of the fraction of instances in the data set belonging to that node, and no heuristic hard decision process is required. This is explored further in Section 5.4.

5.3.4.5 Empirical Performance

In regard to siblings and inclusive training approaches, neither has emerged a clear winner in terms of empirical performance on real data sets [81]. With the difference being a larger set of negative examples from elsewhere in the tree (in the inclusive case), and the scarcity of labels at large tree depths, it may simply be the case that the improvement in individual classifier performance is sometimes more valuable than satisfying the conditional probability assumption more correctly, and depends on the data set being used.

Prediction via multiplication is already a popular approach, and can perform competitively when MLNP is applied, to force prediction to leaf nodes (although results are biased away from areas of the tree with high class depth). Without MLNP, the node with the highest probability will always be one of the direct children of the root node. It is interesting to note that the geometric mean is often used successfully [15], and can be considered to as a method of normalising the posterior probability at each node to account for path length.
5.4 Empirical Application: Methodology

The intention of this study is primarily to establish a more principled approach to hierarchical classification. As such, a widespread comparison of results on a range of data sets was not conducted. A single data set (BENTHOZ-2015) is used to illustrate the practical implementation of the approach, and provide some empirical results in comparison to other commonly used approaches.

5.4.1 Hierarchical Classification with BENTHOZ-2015

The BENTHOZ-2015 data set described in Chapter 4 was used to explore empirical results in a large, real world hierarchical classification problem. It consists of approximately 10,000 images that have been annotated with around 400,000 example points labelled by marine scientists. The labels conform to the CATAMI hierarchy described in [2], a jointly agreed scientific labelling taxonomy agreed upon by a number of marine science groups within Australia. The data set consists of labels from four separate research groups, scattered around the Australian coast (including Western Australia, Tasmania, New South Wales and Queensland); shown again in Figure 5.4.

In [14], an initial exploration of hierarchical classification using the Tasmanian data and the CATAMI hierarchy was performed. Approximately 50,000 point labels from The Tasmania 2008 annotation set (from [12]) were mapped to 20 nodes in the CATAMI hierarchy. In this study using BENTHOZ-2015, 148 classes in the CATAMI hierarchy have positive examples (63 of which had sufficient data to train a meaningful local classifier); shown in Figure 5.5.

Compared to the existing benthic image classification literature (using binary and flat classification), this approach permits a single classifier to be used for a range of
5.4 Empirical Application: Methodology

**Figure 5.4:** Location of autonomous underwater vehicle campaigns in the BENTHOZ-2015 data set
Figure 5.5: The CATAMI hierarchy, including only data present in BENTHOZ-2015. Numbers and edge width indicate cumulative class frequencies.
problems (such as detecting sand, kelp or macro algae), by predicting the likelihood of membership of any node in the tree. Further, it enables a variety of data sets to be easily combined into a single training set, by mapping classes to the CATAMI hierarchy. This adds much greater geographic and temporal diversity, and larger numbers of labels.

For application in field deployed AUVs, the move from flat to hierarchical classification is of crucial importance. With hierarchical classification, it is not necessary to train a new classifier for each deployment in a different geographical location, or for each set of classes of interest to the science goals of the mission. Instead, the far more practical scenario of reusing the same universal classifier on a wide variety of problems becomes possible.

5.4.2 Training Local Classifiers

5.4.2.1 Feature Extraction

The greyscale uniform local binary pattern image descriptor was used as the feature input to the classifiers, on a patch centred on the manually labelled pixel. As in [70], three scales are used (at 1, 2 and 3 pixel radius), with 8, 16 and 24 points at each scale. Concatenating the LBP histograms at each scale results in a feature vector of 54 elements. Based on experience in [12, 14], a $127 \times 127$ pixel patch was used. Although [12] found improved performance on the Tasmanian portion of BENTHOZ-2015 when the LBP was computed on red, green and blue channels, [61] argues that with varying illumination, greyscale only results are superior, so is a more conservative choice when using a data set with increased heterogeneity in image appearance. Further, much of the computational cost of predicting results lies in the LBP calculation, so reducing to a single channel improves the feasibility of
running the classifier live on an AUV with limited computational resources (initial empirical testing indicated that computing half a dozen image patches every few seconds was feasible on a vehicle such as *Sirius*).

### 5.4.2.2 Classifier Training

Random Forests have received much attention since their introduction in 2001 [19], as a general purpose supervised learning algorithm capable of excellent performance across a wide range of problems\(^5\). As such, it is a sensible choice for the node classifiers in a local classifier per node framework. In empirical testing, local node classifiers on the BENTHOZ-2015 data set were found to have higher performance (typically by several percentage points) for most classes when using Random Forests, compared to L2 regularised logistic regression.

The *scikit-learn* implementation of Random Forests was used, with the number of decision trees set to 100 based on computational constraints. “Auto” class weighting was used to penalise mistakes on the minor class proportionally more severely, greatly improving performance over the standard equal cost case. In cases where there were less than 50 positive and 50 negative training instances, the local classifier was replaced by a constant output representing the observed frequency of that class in training set, in keeping with the Bayesian Network assumption.

The data set was divided by randomly selecting a fixed fraction of images from each AUV deployment, and using all annotations applied to that image. 70% of the images were used for training, 15% for calibration, and 15% for testing. Note that the BENTHOZ-2015 data set describes a non-public test set, which was not used at any stage during this work.

---

\(^5\)In a recent study [41], Random Forests were found to be the best of 179 classifiers on 90% of 121 different machine learning databases.
Two distinct sets of local classifiers was trained, using both the *siblings* and *inclusive* approach. The hypothesis was that the *inclusive* approach would require a more dramatic recalibration (on *siblings* data), and therefore demonstrate the benefits of correct classification more distinctly.

### 5.4.2.3 Local Classifier Performance

Prior to combining in a hierarchical scheme, the performance of each classifier was tested on the *siblings* style training approach. Performance was evaluated by examining the ROC, confusion table, reliability plot and probability distribution. Further, the precision of the classifiers (thresholded to balance sensitivity and specificity) was plotted with the class imbalance and sensitivity/specificity (Figure 5.6), for comparison against the theoretical plot in Figure 5.3. All performance metrics and plots show results on the 15% test set.

### 5.4.2.4 Local Classifier Calibration

The probabilities produced by Random Forests classifiers have been found to have varying calibration quality when used directly[69]. Further, the *inclusive* trained set of local classifiers requires recalibrating on *siblings* data in order to meet the conditional probability assumption. Calibration was performed using *scikit-learn*’s calibration module, using isotonic regression after it was observed that many of the reliability plots did not appear to have the sigmoidal shape required for Platt calibration to perform well.

Calibrated classifiers were rolled back to the original, uncalibrated form where either the ROC AUC on the calibration set reduced by at least 1%, or the calibration score (from the decomposed Brier score) became worse.
5.4.3 Prediction Frameworks

The assertion in this study is that the most correct approach is to calibrate the local classifier outputs (in order to best satisfy the conditional probability assumption), and then to perform inference by multiplying ancestor paths of probability.

There are two main hypotheses related to this assertion. Firstly, that performing calibration on the nodes will improve the quality of the probabilistic output of the classifiers, rendering the Bayesian Network assumptions more valid. Secondly, that the abundance estimates produced by the calibrated Bayesian Network will be superior to those produced by the other heuristic approaches.

5.4.3.1 Bayesian Network

An abundance estimate in ecology aims to estimate the observed frequency of a class from a given geographic area. Bayesian Network inference (via multiplication) is performed, and the average inferred probability for each node is used as the abundance estimate for each class. It is not possible to perform hard classification of an example in a Bayesian Network without some heuristic approach, as deeper nodes in the tree will necessarily have lower probabilities than the highest ones, so only top layer nodes would ever be chosen via the MAP estimate. Further, an MLNP approach would be inappropriate, given the variable tree depth. However, the modified hierarchical Brier score can still be computed. The geometric mean approach (below) could be considered as a heuristic normalisation of the Bayesian inference to account for this difference and permit hard, single path prediction with MLNP.
5.4.3.2 Geometric Mean

Geometric Mean prediction takes the n-th root of the Bayesian Network inferred probability, where n is the node depth in the tree. Hard classification was performed with and without MLNP, and the hard predictions were used for computing both the abundance estimate and the modified hierarchical loss function. In the MLNP case, hard outputs are selected by choosing the leaf node with the greatest aggregated probability. The path length normalisation effect of the geometric mean makes this a much more defensible approach on a tree with variable depth than in the pure Bayesian Network case. An heuristic approach was taken to relaxing the MLNP constraint; leaf nodes were redefined by dynamically pruning every set of child nodes where all of the siblings gave predictions lower than a probability threshold. That threshold was chosen by optimising the modified hierarchical loss function on a separate calibration data set. Hard predictions were converted to abundance estimates by taking the fraction of test set examples with a positive prediction for any given node.

5.4.3.3 Greedy Walk

A greedy walk necessarily provides hard prediction outputs, with no notion of a probabilistic estimate. The greedy walk was performed up to a probability threshold (i.e. the walk stops when all child nodes have a probabilistic classifier output less than a threshold). The threshold was selected in the same manner as that for the geometric mean. Full depth MLNP prediction was also performed (equivalent to setting a zero threshold).
5.4.4 Local Classifier Performance

The accuracy of the probabilistic estimates of individual classifiers is measured using the calibration score from the two-component decomposition of the Brier score, evaluated over the unseen test set.

5.4.5 Hierarchical Performance

The modified hierarchical f1-score was introduced in Chapter 3[14]. Here, an additional hierarchical performance metric is defined: the modified hierarchical Brier score. Based on the Brier score for binary classification, the hierarchical version presented here computes the mean squared difference between the true label and probabilistic prediction over all classes and examples. The same modification as with the modified hierarchical f1-score is used to remove penalties from the descendants of the ground truth node. The score is defined as:

$$B_{modified} = \frac{1}{NH - U} \sum_{i=1}^{N} \sum_{j=1}^{H} \frac{1}{N} (\hat{y}_{i,j} - y_{i,j})^2$$

(5.1)

$N$ is the number of examples in the test set, $H$ is the number of classes in the hierarchy, and $U$ is the count of cases where the true label $y_{i,j}$ for example $i$ is unknown for class $j$ due to partial depth labelling.

The difference in interpretation between the modified hierarchical Brier and f1 scores is analogous to the difference between the standard loss, and the f1-score. The former simply counts the number of errors that occur, where the latter penalises differences between precision and recall (so is particularly suited to imbalanced data). While it would appear that the f1-score is therefore more desirable, it can only be reasonably
computed on binary classifier outputs, whereas the Brier score applies to both binary and probabilistic output.

5.4.6 Abundance Estimates

Lastly, the overall class abundance estimates were compared for the various hierarchical prediction cases. Due to the fact that true labels were often not specified to leaf node level, it was not possible to compute a simple ground truth for comparison (e.g. the true abundance of *Ecklonia radiata* is uncertain, as some unknown portion of the *Macroalgae* or *Biota* classes may be *Ecklonia radiata*). This results in maximum and minimum bounds for the true abundance for each class. The error metric used to evaluate abundance accuracy was taken as the distance of the estimate from the nearest bound (i.e. for under prediction, the distance below the minimum bound, and over prediction, the distance above the maximum bound). Any abundance estimate falling within the bounds was considered to have zero error.

Abundance estimates (predicted class frequencies) were computed for each class in the hierarchy, using the various prediction frameworks. For those producing hard predictions, the abundance was taken as the fraction of examples which were predicted positive for that class, after the prediction strategy had been applied (i.e. only single path prediction was permitted). For the soft predictions produced by the Bayesian network, the mean predicted probability (after multiplying by ancestor probabilities) was used as the abundance estimate.
5.5 Empirical Application: Results and Discussion

5.5.1 Local Classifiers

5.5.1.1 Siblings classifiers

Of the 148 nodes with examples in the class hierarchy, 63 had sufficient training instances (at least 50 of each class) and differing visual appearance to produce un-calibrated classifiers with an ROC AUC score greater than 50% on the validation set. Figure 5.6 was created as an empirical version of Figure 5.3, by finding the threshold for each classifier that balanced sensitivity and specificity equally.

The reliability plots (upper left plots in examples in Figure 5.7) show that good performance does not always result in good calibration (as in the case of the *Macroalgae* classifier in 5.7b). Given random forests is one of the more commonly used classification algorithms, it is important to note the sometimes poor calibration of the...
probabilities when used directly. Figure 5.10 shows a comparison before and after calibration. 42 of the classifiers met the criteria to use the calibrated version (no appreciable worsening of AUC or calibration score).

### 5.5.1.2 Inclusive classifiers

The process used for the *siblings* classifiers was repeated, training on the *inclusive* data set (with a larger set of negative labels from elsewhere in the tree). It is important to note, however, that both the calibration step, and the performance testing were done using the same *siblings* data set.

Figure 5.8 shows that in general, results for the *inclusive* classifier on the *siblings* problem were on the whole substantially worse. While it may appear that the *inclusive* approach should immediately be discounted, it is important to consider that the hierarchical classification task involves every single instance being tested against every classifier - so the larger variety of negative labels is expected to improve hierarchical performance in some sense.

Observing the examples of specific classifiers in Figure 5.9, it can be seen that the calibration of the *inclusive* classifiers is substantially worse (by an order of magnitude). 50 of the *inclusive* classifiers met the criteria to use the isotonic calibration step. This difference is to be expected, as the training set was chosen according to a different policy than the tet set. A rare class deep in the hierarchy with no siblings would result in a fixed 100% prediction on the *siblings* problem, and a classifier trained on a small number of negative examples, and many positive ones under the *inclusive* scheme. After correction (using the *siblings* calibration set), calibration performance on the unseen test set is comparable to that seen in the *siblings* trained classifiers. This result implies that classifiers trained using either training set approach, with the appropriate correction, can be used to represent the conditional
Figure 5.7: Example reliability plots, prediction probability distributions, and ROC curves using the siblings training set definition. The top row reliability plot shows reliability of the calibration set (green) and the test set (blue). The second row of each sub-figure shows the test set results after calibration.
5.5 Empirical Application: Results and Discussion

Figure 5.8: Empirical variation of precision with class imbalance for the inclusive trained classifiers, after the threshold is applied that balances sensitivity and specificity. Dot radius is proportional to AUC score (shifted for zero radius at AUC=0.5), and colour represents precision.

probability. It would not be unreasonable to suggest that any choice of training data for the classifier can be corrected post hoc with an appropriate calibration set (which is likely to be much smaller than the amount of data required to train a strongly performing classifier).

5.5.1.3 Performance Comparison and Calibration

There were 35 classes for which a random forest was trained, and separate calibration was performed on both the siblings and inclusive problems. In order to test whether the calibration score was typically improved by the correction (measured on the test set), the non-parametric paired, one-sided Mann-Whitney-Wilcoxon was applied.

In Figure 5.10, it can be seen that the initial calibration of the inclusive classifiers was far worse than for siblings (at $p = 1 \times 10^{-10}$). Calibration made an enormous difference on the inclusive classifiers (at $p = 6 \times 10^{-11}$), which was expected. The siblings result also benefited substantially (at $p = 2 \times 10^{-4}$), which is noteworthy.
Figure 5.9: Example reliability plots, prediction probability distributions, and ROC curves for the inclusive classifiers. Note that the results are shown tested against the siblings data set.
5.5 Empirical Application: Results and Discussion

Figure 5.10: Performance comparison of local node classifiers before and after calibration, measured on the calibration score from the Brier two-component decomposition. Lower scores indicate superior calibration.

(a) Calibration Score - Improvement with calibration correction

(b) Calibration Score - Comparison between siblings and inclusive

given that this step is never applied (or at least described) in the hierarchical classification literature. Finally, there was no evidence to suggest that post-calibration results of the two different approaches differed (at $p = 0.2$ on a two-sided test). The isotonic regression calibration step was sufficient to completely correct results for the inclusive approach, such that the probabilities reasonably fit the assumption of representing the conditional probability given the parent class.

5.5.2 Hierarchical Performance

Figure 5.11 shows the results for the Bayesian Network prediction approach on both siblings and inclusive trained classifier hierarchies, using the modified hierarchical Brier score. Prior to calibration correction, the siblings classifier hierarchy outperforms the inclusive one by a large margin. After calibration for all classifiers is
corrected to conform to the *siblings* data set, there is a slight improvement in the *siblings* score, and a substantial one for the *inclusive* score (to the extent that the *siblings* and *inclusive* approaches perform similarly). This fits well with the above results on individual classifiers, and suggests that appropriate calibration is crucial to getting the best performance with the Bayesian Network prediction approach of multiplying probabilities. Further, it suggests that despite concerns such as in [15], it is possible to make a probabilistic interpretation for *inclusive* trained classifiers, so long as calibration is performed.

The three other approaches tested (greedy walk with MLNP, greedy walk with 50% threshold, and geometric mean with MLNP) are all hard classification techniques, and constrained to enforce a single path prediction in the hierarchy. Direct comparison to the probabilistic predictions of the Bayesian Network approach would therefore not be valid (although a Brier score can technically be computed). Instead, their performance is assessed as a group using the modified hierarchical f1-score described earlier, which is better suited to hard classification, and is commonly used to assess performance in the hierarchical classification literature.

The greedy walk pruned to a 50% probability threshold performs best in both the *inclusive* and *siblings* cases. It is unsurprising that this outperforms the MLNP case, where positive predictions were forced at the deepest (and most inaccurate) levels of the hierarchy. Examining the scores in more detail, it can be seen that the pruning reduces hierarchical recall by a small amount (approximately 1-2%), but improves hierarchical precision by a larger amount (4-5%). This asymmetry is expected as the hierarchical precision is negatively impacted by the large number of false positives caused by forcing a mistake high up in the tree to predict all the way to leaf node. Similarly, the geometric mean is forced to consider the full set of probabilities from root to leaf node, so may perform most poorly because of the inclusion of forced
5.5 Empirical Application: Results and Discussion

(a) Bayesian Network approaches, using modified hierarchical Brier score
(b) Hard classification approaches, using modified hierarchical f1-score

Figure 5.11: Hierarchical performance of four commonly used prediction strategies, with and without local classifier calibration. Note that for the modified hierarchical Brier score (a loss), lower is better; for the modified hierarchical f1-score, higher is better.

Inclusion of poorly performing classifiers deep in the hierarchy with every decision.

Calibration appears to improve performance noticeably for both greedy walk approaches, regardless of training set definition. This could be interpreted as indicating that correctly calibrated probabilities permit a more valid comparison between child node probabilities, when the greedy walk algorithm is attempting to choose the most likely child node at each step. While the change with calibration in the geometric mean case is in the other direction, it might be considered less important as the change is both smaller, and on the technique that is already performing the worst.

5.5.3 Abundance Estimates

Figure 5.12 shows the abundance estimates for the top three layers of the class hierarchy, for the siblings classifier hierarchy. The black error bounds denote the
minimum and maximum abundance range for the ground truth data, where the ambiguity results from partial depth labels (e.g. identified as “Macroalgae”). It is likely that the true value typically lies close to the lower bounds, as the true class can only follow a single path from the most specific label. Put another way, if the unknown true abundance of one class was the maximum error bound, all the siblings of that class would be constrained to the minimum error bound.

The calibrated Bayesian Network prediction strategy clearly produces the best abundance estimates. In cases where there was little or no ambiguity, the Bayesian Network typically produced estimates much closer to the known true abundance. Table 5.3 shows the mean absolute error across all classes for Bayesian Networks as an order of magnitude lower than Geometric Mean, and Greedy Walk approximately double the error of Geometric Mean. In cases with a wider abundance range (where zero error is counted), it typically estimated at the lower end within the target range.

For the other prediction frameworks, the abundance estimates were in general poor. As an example, the Substrate: Consolidated (hard) class near the bottom of Figure 5.12 is essentially rocky areas on the sea floor. The Bayesian Network estimate was very close to the true value around 2.5% abundance, however the other estimates relying on hard classification were close to 0% presence. Worse, the greedy walk approaches both drastically overestimated the abundance of biota at the top level, even after calibration. As expected, the GW.MLNP case (where prediction was forced to leaf node) produces substantially higher estimates for classes deeper in the hierarchy than those cutting off at a threshold. This under-prediction of rarer classes occurs frequently in the heuristic methods (of order 10 classes underestimated for the Bayesian Networks, and 100 for the Geometric Mean and Greedy Walks), and is a significant disadvantage when applying these methods to hierarchical classification problems for which accurate abundance estimates are important, such as benthic
Table 5.3: Aggregated error for each strategy with/without calibration. Calculations are performed across all classes in the hierarchy. Mean $|e|$ is the mean absolute error (where the absolute error is the distance outside the true class bound, as described in Section 5.4.6). $N$ Underestimated and $N$ Overestimated are the number of classes where the estimate fall below or above the known range, respectively.

Calibration improves the mean absolute error for all cases except the Sibling Bayesian Network, and Inclusive Geometric Mean approaches. The increased error with calibration of the Bayesian network can be attributed to very slight adjustments in the estimates for some of the most abundant classes *Biota* (75.2% to 74.6%), *Physical and Substrate* (24.7% to 25.4%) and *Substrate Unconsolidated (Soft)* (22.3% to 23.0%). With these four classes excluded, calibration improves the mean absolute error in all cases. By contrast, the Greedy Walks are at least 5% (absolute error) away from the true value.

The inclusive hierarchy of classifiers saw a dramatic difference in abundances for many of the classes after calibration, most notable in the *Macroleague* class and its ecology.
### Figure 5.12: Siblings abundance estimates for classes in the first three layers of the tree, listed in breadth first search order. For each class, the min/max abundance range is displayed as error bars, defined by assuming all ambiguous labels are either negative or positive on the test set. “BN”, “GW” and GM” refer to Bayesian Network inference (i.e. multiplication), Greedy Walk, and Geometric Mean as prediction strategies. “MLNP” indicates when predictions have been forced to leaf node, rather than using a cutoff threshold.
children. After calibration, the results were similar to the siblings case. This is reflected in the mean absolute error from Table 5.3, which is similar in the post-calibrated inclusive and siblings cases, but approximately 20 times greater for the uncalibrated inclusive approach. The post-calibration error for siblings and inclusive Greedy Walks and Geometric Mean is also comparable.

A point of interest can be made regarding the Substrate class and its descendants. As can be seen in Figure 5.5, Substrate is the only child of the Physical classifier. For the siblings case, this means there were no negative examples to train a classifier, so a fixed prediction was used (entirely appropriately): \( P(\text{Substrate}|\text{Physical}) = 1 \). In the inclusive case, however, the positive and negative training examples for the two classifiers were identical, resulting in essentially the same classifier being trained twice, connected one after the other. In the case of greedy walk, this makes no difference. For the Bayesian Network classifier, this effectively results in greatly reducing the abundance estimate for the descendant classes. In the limit of a long chain of such classes, there is no impact on the siblings case, whereas the inclusive predictions reduce asymptotically to 0%. Calibration, however, is able to protect the inclusive approach from this effect.

### 5.6 Conclusion

Hierarchical classification is a distinct type of classification problem, alongside binary and flat multi-class classification. There are a wide range of approaches to solving it, split largely into those that solve it as a single global problem (as is the case with structured SVMs), and those that aggregate the predictions of a set of classifiers trained on sub-problems. The *local classifier per node* approach involves training a distinct classifier to predict membership of each node in the class hierarchy. Even
Figure 5.13: Inclusive abundance estimates, plotted in the same way as Figure 5.12.
within the *local classifier per node* approach, there are fundamental decisions to be made around the training set to be used, and the method of aggregating outputs. A principled approach using the theory of Bayesian Networks and conditional probability distributions was proposed, and a comparison of empirical results performed on BENTHOZ-2015, a large hierarchical data set of scientifically annotated marine images [13]. The empirical results demonstrate vastly superior performance for the Bayesian Network in estimating class abundance. Further, by utilising a sound theoretical framework, principled grounds were available for making a number of decisions. To satisfy the conditional dependence assumption, *siblings* rather than *inclusive* training sets were initially used to train nodes, and local classifier calibration was performed. A novel concept of re-calibrating an *inclusive* hierarchy of classifiers using *siblings* data was also introduced, in order to satisfy the conditional probability assumption with a classifier built on any training set choice.

From results on only a single data set, it would be inappropriate to argue that the principled Bayesian Network provides universally superior performance. However, there was evidence for a number of useful properties of the approach. Bayesian Networks obtained by far the best abundance estimates, particularly when calibration was performed, with the heuristic hard classification approaches consistently underestimating rare classes (likely due to the issue described in [10]), and even performing poorly on the most abundant classes. The performance of the *inclusive* approach was improved by a large margin to the same level as *siblings* after calibration correction using isotonic regression. Given the literature suggests that *inclusive* is often superior on data sets even without calibration, this suggests the potential for further improvement. Calibration was shown to improve performance on the greedy walk strategy, so may prove useful even when a hard, single path prediction approach is required.
Bayesian networks can be used without resorting to decisions based on empirical performance, and have the potential to be used in other more complex configurations, such as in a DAG class hierarchy (with multiple parents). The only drawback of the Bayesian Network approach is the lack of ability to make a hard prediction of a single path prediction class from the hierarchy. However, it is ideal for answering the question of which of a set of examples belong to a specific class. This combination of properties makes the Bayesian Network approach ideally suited to ecological (and other) applications where abundance estimates must be unbiased, and where an existing set of classifiers may be “tuned” via calibration to predict more accurately on a new data set from e.g. a different geographical region.

Much of the hierarchical classification literature has focussed on evaluating the performance of a wide range of approaches on given data sets, with inconclusive results. In this work, the link to Bayesian Networks was explicitly made, and the argument put that the benefit of the sound theoretical framework can simplify the problem space, assist understanding, and provide excellent empirical performance results. For any application where the abundance of each class is important, the unthresholded, calibrated Bayesian Network is the most promising approach. Use of the heuristic approaches should be restricted to cases where it is strictly necessary to produce a single, hard output, and the other disadvantages are not of concern.
6 Hidden Node Bayesian Networks for Hierarchical Classification

In Chapter 5, a relatively simple application of Bayesian Network theory was applied to explain and refine a common existing approach to hierarchical classification. A natural question arises as to whether there are benefits in extending the use of Bayesian Networks to more complex approaches, using additional properties of Bayesian Networks, and more complex configurations. This chapter investigates one such configuration, which is particularly suited to applications in robotics and environmental monitoring.

6.1 Background: Hierarchical Bayesian Aggregation

The implicit Bayesian Network interpretation of the local classifier per node hierarchical classifier introduced in Chapter 5 relies heavily on an assumption that the probabilistic output of a local classifier directly represents the conditional probability of class membership, given the parent class. In [8], a different Bayesian Network formulation for the problem was introduced, which is described by the authors as Hierarchical Bayesian Aggregation. Instead of making the implicit conditional probability assumption, the class hierarchy is augmented with a new set of child nodes.
The original, binary nodes represent whether an example is truly a member of that class. Each original node has a child node that represents the output of local classifier (in that case, a kNN classifier is trained for each node in the class hierarchy). An example of this structure using benthic classes is shown in Figure 6.1. Prediction is performed by Bayesian inference, computing the marginal probability of membership of each of the hidden true class nodes. The hierarchy is enforced through the conditional probability tables of the hidden nodes (stating 0% likelihood of membership given an example is not a member of the parent node). An extensive description of the approach can be found in the PhD thesis publication [7], demonstrating successful empirical results on gene function prediction, geometric shape matching and music genre classification.

The performance in [7] was assessed primarily by comparing ROC AUC scores for the probability produced by Bayesian inference for each node, with that produced by each of the individual classifiers. This assumes a multi-label hierarchical problem (in which no single forced prediction is made of the most likely node). It is important to note that no comparison is made with any other hierarchical classification framework (including the commonly used local classifier per node formulation described in Chapter 5), nor on an hierarchical performance metric.

A key advantage of the hidden node formulation is that the classifier outputs are no longer required to directly represent conditional probabilities. Although [8] uses the binary classifier outputs, this was extended to a Gaussian Bayesian Network based on unthresholded SVM outputs in [7].
6.2 Hidden Node Bayesian Networks

To contrast this approach with the implicit Bayesian Networks of Chapter 5, we use the term Hidden Node Bayesian Network rather than Hierarchical Bayesian Aggregation. The defining feature between the two approaches is the separation of the true node definition from a classifier prediction, whereas both approaches involve hierarchies, Bayesian networks and aggregation. In fact, [7] notes that the Bayesian Aggregation (Hidden Node Bayesian Network) model converges to a hidden Markov model in the case where the class hierarchy is a single chain, so our terminology appears appropriate. This work makes several novel contributions extending beyond [7]:

- Multiple, distinct local classifiers are introduced for each hidden node, rather than the single classifier per node in [7]. This affords a number of signifi-
cant benefits for environmental monitoring robotics: Redundancy to missing classifier outputs (e.g. sensor failure), maintaining the capacity for intuitive understanding by scientists (e.g. separate observable classifier outputs based on image texture and other sensor data), ability to tune conditional probabilities to suit new environments, and ability to incrementally add new classifiers (based on e.g. 3D structural information from a stereo camera, or expert knowledge) without altering or retraining the existing classifiers.

- A comparison of the hidden node structure against the implicit Bayesian Network structure (and other heuristic hierarchical classification approaches) on the same data set, using the same local classifiers, and a hierarchical (rather than per-class) performance metric. While the hidden node structure makes fewer assumptions and is more elegant than these other approaches, it is valuable to compare them empirically.

- A comparison of results for inclusive vs siblings training, as well as calibrated vs uncalibrated classifiers.

6.3 Methodology

6.3.1 Hidden Node Bayesian Network Structure

As described above, the same fundamental structure for the Hidden Node Bayesian Network is used as in [7]. The same sets of local classifiers from Chapter 5 were reused without alteration, to ensure a fair comparison against the implicit Bayesian Network approach (comparing calibrated, uncalibrated, siblings and inclusive). Given the conditional probability assumption is no longer required to hold, the hypothesis was that calibration would no longer play a significant role in the hierarchical per-
formance. As a whole, this testing compares the hidden node and implicit Bayesian Network approaches using the modified hierarchical Brier score defined in Chapter 5.

### 6.3.2 Training Local Classifiers

AUVs typically have a range of different sensor types, including greyscale cameras, RGB cameras, stereo cameras (for 3D features), hyperspectral sensors, multibeam (acoustic) sensors, depth, salinity, chlorophyll and other environmental sensors. While one approach would be to include all available data as inputs in a single set of local classifiers, this reduces the interpretability of the model, in the sense that it is more difficult for a marine scientist or field operator to understand why a decision is being made (e.g. errors being caused by a malfunctioning depth sensor). Further, different AUVs have different sensor configurations, and sensors may be unable to operate for part or all of a deployment. Using separate local classifiers for different sensor modalities in a Bayesian network is therefore particularly advantageous.

The set of 148 local node classifiers described in Chapter 5 essentially predict the probability of class membership, based on the texture of the greyscale image in the immediate vicinity of the labelled point. Here, an additional set of 148 classifiers was trained to perform the same task (*siblings* prediction of each class in the tree), based on other sensor data only. To illustrate this functionality, only the depth variable was used. However, it would be possible to include others such as salinity, temperature, latitude, longitude, and 3D stereo features. The same classifier algorithm was used (random forests with 100 trees), and the same training, calibration and test set split. It should be noted that the depth based classifier will be highly biased to the distribution of classes found as part of BENTHOZ-2015 [13]. This does not impact the validity of the conclusions drawn here, but implies that calibration to
expected depth distributions in a new data set is of particular importance. Perhaps the best way to avoid excessive bias in a depth based classifier would be for a human expert to simply define which species and classes are plausibly found at each depth. This would, however, require significant manual effort, and add little to the analysis performed here. Instead, the sensor classifiers were trained using an RBF kernel Support Vector Machine, with hand tuned parameters to ensure smooth depth vs probability curves. With the siblings approach, this was necessary to avoid overfitting to the data set. Pockets of particular classes were frequently found at tight depth ranges on a given mission, and may not have been found within other missions in the data set overall. The advantage of this approach is that the depth curves can be observed (or altered) by a marine scientist, and checked for inappropriate behaviour (by asking e.g. which type of macro algae is most common at each depth?).

6.3.2.1 Network Structure

The conditional probability tables of the hidden (true class) nodes take the same intuitive meaning as in Figure 5.2. When the parent class is negative, the child must also be negative, so the first column enforces the hierarchical structure, and is independent of the data set. When the parent class is positive, the table simply represents what fraction of parent node instances belong to the child (e.g. “What fraction of Biota is also algae?”). Where Chapter 5 uses the calibrated classifier predictions as a proxy for this, here it is simply computed from the fractions of each type in the training set.

The conditional probability tables of the observed (classifier prediction) nodes also take on a clear intuitive meaning. In the binary classification scenario, the conditional probability table is closely related to the classifier’s confusion matrix. The
sensitivity \( \frac{TP}{TP + FN} \) is the proportion of true class positives that are predicted to be true. This is exactly the same as the conditional probability of a child given the parent (see Table 6.1). Similarly, the specificity \( \frac{TN}{TN + FP} \) is the conditional probability of not observing the child, given the parent was not observed. This concept extends readily (although without the convenient terminology) when the classifier output is discretised to more than two levels.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>True Class</th>
<th>True Class Negative</th>
<th>True Class Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted Negative</td>
<td>specificity</td>
<td>false negative rate</td>
<td>sensitivity</td>
</tr>
<tr>
<td>Predicted Positive</td>
<td>false positive rate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: The conditional probability table of an observed classifier prediction node corresponds to the performance metrics of the classifier.

Taken together, observation of these two types of conditional probability table gives this framework a high degree of transparency, and neatly separates two distinct concepts:

- The hidden node (true class) tables show a simple prior for how often a class is seen, given its parent;
- the observed node (classifier prediction) table shows the discriminative power of each classifier.

As well as being useful in gaining an understanding of how the classifier system operates, it is also possible to directly manipulate these tables to achieve a desired effect, without retraining any classifiers.

In Bayesian Networks, there is some flexibility of definition with the directed edges. While [7] makes an argument for reversing the direction (and achieves slightly increased performance in some situations), this neat interpretability is broken. In the reverse case, it is necessary to model the likelihood of the parent class given the child, which is somewhat counter-intuitive. The investigation performed here was
restricted to discrete Bayesian Networks for similar reasons; the ability to simply read off the classifier performance in a discrete network can aid a marine scientist in understanding to what extent they should trust the behaviour of the network.

6.3.3 Training the Network

A discrete Bayesian Network is defined by two things: The structure of the tree (nodes and edges), and the conditional probability tables. The restricted CATAMI hierarchy of 148 nodes and associated classifiers was used without alteration from Chapter 5. Both the hidden and observed node tables were defined based on a combined table of true labels and classifier predictions for the whole training set.

6.3.3.1 Discretisation

The hidden nodes are inherently binary; an example either is, or is not a member of a class. The probabilistic classifier outputs, however, are a continuous value in the range $[0,1]$. As described above, discretisation is necessary to retain intuitive understanding of the network parameters. After empirical testing with various schemes (fixed width bins, equal frequency bins, and supervised binning techniques such as ChiMerge [88], which merges neighbouring bins until a chi-squared test finds evidence of difference in the output, and MDLP [4], which uses the minimum description length to identify cut points), it was found that the fixed width binning approach was likely to give the best performance (when a single scheme was applied to all classifier outputs). This is convenient, as it permits the discretised level for any classifier output to be immediately interpreted as a probability range (e.g. 4-bin discretisation results in quartile bin edges at 0%, 25%, 50%, 75% and 100% for all nodes). For each of the hidden node networks, the number of bins was selected for maximum performance of the modified hierarchical Brier score on the test set.
6.3 Methodology

6.3.3.2 Computing conditional probability tables

The \textit{bnlearn} package in R\cite{bnlearn} was used to handle the Bayesian Network. Two complications in the data prevented the use of the default method that computes conditional probability tables from a table of data.

1. Many classifier nodes did not have all of the discretised levels observed in the data set (primarily at nodes deep in the tree with few training instances, and those with constant predictors).

2. The true labels include a substantial fraction of unknown values, where a label has been provided at some non-leaf node in the tree.

The first constraint was dealt with by adding a weak uniform prior. This was achieved very simply by appending a list of synthetic dummy examples to those from the training set, before computing the contingency table. One example for each valid combination was added (i.e. for classifier node tables, with \texttt{parent} = 1, add one entry for each discrete output level; with \texttt{parent} = 0, add a single entry at the lowest discrete level. For true class nodes, just three entries were required for \texttt{(node|parent)}: (0, 0), (0, 1) and (1, 1). For large amounts of training data, these dummy examples have almost no impact. For nodes with very few positive examples, however, it ensures that probabilities of 100\% or 0\% do not appear inappropriately. By never including the forbidden case of \texttt{child} = 1|\texttt{parent} = 0, the conditional probability tables still enforce a 0\% chance of observing a child node, given the example does not belong to the parent node. This unusual use of extreme probabilities is acceptable, given the point is to strictly enforce a definitional requirement of the problem.

The second constraint was simply a limitation of the \textit{bnlearn} package; in our method, the contingency table of raw counts was constructed ignoring missing values, and then converted to a conditional probability distribution.
Chapter 6 Hidden Node Bayesian Networks for Hierarchical Classification

The same 15% calibration set used to calibrate the classifiers was used to compute the class frequencies in the conditional probability table. Given the relationship between sensitivity, specificity and the conditional probabilities, the training data set would be likely to provide overly optimistic estimates.

6.3.4 Computing Inference

For the implicit Bayesian networks, exact inference is trivial; probabilities of a node and it’s ancestors are multiplied to find the marginal probability of class membership. The \textit{gRain} package in R was used to perform exact inference on the Hidden Node Bayesian Networks. For each example in the test set, the discretised classifier observations were supplied to the network as evidence, and the marginal probability was computed using \textit{gRain} for each of the true class nodes.

6.3.5 Network Configurations

Five different network configurations were tested:

1. Implicit Bayesian Network with texture classifiers.
2. Implicit Bayesian Network with sensor classifiers.
3. Hidden Node Bayesian Network with texture classifiers.
4. Hidden Node Bayesian Network with sensor classifiers.
5. Hidden Node Bayesian Network with both texture and sensor classifiers (as separate children).

Cases 1. and 2. are used as comparison points with the technique described in Chapter 5. Cases 3. and 4. demonstrate an approach similar to [7], and Case 5. demonstrates the extension to multiple distinct classifiers. The hypothesis is that
the combined hidden node approach in Case 5. can provide superior performance to either 3. or 4. (and will revert to the behaviour of 3. or 4. if positional or texture classifier outputs are unavailable, e.g. due to sensor malfunction).

6.3.6 Comparing Performance

6.3.6.1 Hierarchical Performance

In Chapter 5, the metrics used to evaluate performance were the modified hierarchical Brier score, and f1-score. As previously discussed, the Brier score is more suited to probabilistic predictions. Both the hidden node and implicit Bayesian Networks produce probabilistic output, however it should be noted that the hidden node network is only capable of producing a discrete set of probabilities, based on the number of bins in the discretisation.

In addition to the five networks described above, performance was also computed on the hidden node networks with no classifier observations supplied as evidence. This results in a constant probabilistic prediction for each node that represents the prior probability of occurrence in the calibration data set. The performance in this case provides a useful benchmark against which to test the relative improvement resulting from the image (greyscale texture) and sensor based classifiers.

The use of a simplistic discretisation scheme for the hidden node Bayesian Networks effectively reduces the information the local classifiers can provide to the network. Given the implicit Bayesian Networks use the direct probabilistic output, a comparison was also performed using a version of the implicit Bayesian Network structure with discretised local node probabilities. Probabilistic values were set to the bin centres (e.g. for 4-bin discretisation, the levels of 0.125, 0.375, 0.625 and 0.875 were used). Inference was performed as normal, by multiplying these discretised proba-
abilities down the ancestor node path. Having both discretised and raw probabilistic
implicit network results provides a double baseline, which may suggest potential
performance increases possible for the hidden node networks if better discretisation,
or continuous probability distributions were used.

6.3.6.2 Abundance Estimates

The accuracy of abundance estimates was assessed in the same manner as in Chapter 5.

6.3.7 Regional Tuning

The local classifiers were built using a training set comprised of a variety of different
regions around Australia. With the hidden node Bayesian Network, the conditional
probability tables also represent relative abundance and classifier performance across
the whole calibration set.

In this section, the decoupling between local classifiers and the conditional prob-
ability tables is exploited. Without altering the local classifiers in any way, the
calibration and test sets were divided into six broad regions (Tasmania, Western
Australia, Batemans, Port Stevens, Solitary Islands, South East Queensland). The
conditional probability tables for both the hidden true class nodes and local classifier
outputs were fit using the region’s calibration set, and inference was then performed
for the same region’s test set. The hypothesis was that the region specific conditional
probability tables are a better reflection of both class abundance and local classi-
fier performance in the region; and that this would reduce the modified hierarchical
Brier score compared to the globally fit model.

The regional tuning simulates adjusting the prior belief of classifier performance and
class abundance, before deploying an AUV in a particular region. These adjustments
could be made based on a number of sources. Options include encoding expert knowledge directly (based on the intended deployment location), class abundances from another source (e.g. rough estimates from satellite data), using data from historical deployments in the area, or rapidly scoring part of the data set at the start of a new campaign.

6.3.8 Sensor Redundancy

In addition to regional tuning, another benefit of the hidden node approach is the redundancy to missing sensor data. This can be simulated by using the best combined approach (with nodes for texture classifiers, sensor classifiers, and true classes), and randomly removing either the texture classifier, sensor classifier, or both inputs from the inference step. On a live AUV deployment, this is the equivalent of the depth sensor and cameras operating intermittently.

6.4 Results and Discussion

6.4.1 Training Local Classifiers

6.4.1.1 Texture Classifiers

The texture based classifiers described in Chapter 5 were re-used without modification.

6.4.1.2 Sensor Classifiers

Based on empirical testing, local classifiers were trained on depth of the sea floor at the point of image capture (spanning 0-120m below the surface). A radial basis
function support vector machine (RBF SVM) was trained to predict the probability of class membership with depth, for every class. Parameters were tuned to $C = 10$ and $\gamma = 10^{-5}$ by observing the depth profile plots (as in Figure 6.2), and ensuring there was sufficient smoothing to remove any unusual peaks or troughs in the probability density across depth. The training data set was trimmed to 1500 positive and 1500 negative examples, randomly selected from the siblings training set for each class. This was done to manage the poor scalability of SVM training time with large sample sizes, and the rebalancing of classes was performed so as not to discard any of the valuable positive instances for rare classes. While the balancing of positive and negative training set sizes is a common practice, it is important to note that it results in a poorly calibrated probability, and consequently poor performance for the uncalibrated sensor classifier cases (Table 6.3).

Due to the nature of the data set, each class has some densely populated data at particular depths (where the AUV happened to operate), and no data at others. In the local classification task, it was common to have, for example, sibling classes with different depth patterns that resulted in a rapidly changing depth vs probability profile. This type of overfitting cannot be accounted for by cross validation, or methods such as random forests, as the sparse distribution of data with depth is fundamentally insufficient to accurately represent what is possible in nature. Taking an RBF SVM with a deliberately extremely wide kernel smoothed these rapid changes, and is intended to represent in a very general sense which classes are more likely to be found at what depth.

The siblings depth profiles can be interpreted as the question “How likely is, for example, a random observation of biota at a particular depth to be a member of the sponges class?” Given the troubled generalisation of the classifiers due to the non-random nature of AUV deployments, it would be entirely appropriate to replace
either set of classifiers with an expert derived curve, defined by a marine biologist based on prior belief of class distributions.

**Figure 6.2:** Example probability with depth profiles, for *siblings* trained classifiers. This represents the probability of class membership, conditioned on the parent.

As a group, the sensor classifiers are far less powerful than the image texture ones, which makes intuitive sense. While a similar number of classifiers had a meaningful AUC (61 above an AUC greater than 50%, as opposed to 63 for the texture classifiers), the implicit Bayesian Network scored much more poorly, as shown in Table 6.3, at 0.01355 for the calibrated *siblings* case, compared to 0.09859 for the
equivalent case using image texture.

Figure 6.3: Siblings local classifier performance plot, as per Figure 5.6

6.4.2 Network Structure

Figure 6.4 depicts the hidden node Bayesian Network with texture classifiers only (primarily to give an overall sense of the scale of the problem). All nodes are discrete, with the true class (id) nodes as binary, and the texture classifier prediction discretised between two and ten levels (with optimal discretisation tuned on the data set).

6.4.3 Training the Network

From empirical testing, it was found that the 4-bin equal width discretisation performed on the calibrated, siblings trained classifiers was the best performing hidden node approach that relied solely on texture classifiers. Example conditional probability tables for both the hidden true class nodes (id_*), and the texture classifier nodes (txt_*) are shown in Table 6.2. Because 4 equal width bins were used for the texture classifier nodes, the interpretation is not quite as simple as the sensitivity, specificity, false positive and true positive rates of the binary case of Table 6.1.
Figure 6.4: Hidden Node Bayesian Network: Structure with separate true class and texture classifier nodes, with some example labels. For node labels, best viewed on computer screen, at high zoom.

However, it is easy to tell at a glance whether the local classifier performed well on the Bayesian Network fitting set (in this case the calibration data set). Similarly, the hidden node tables permit a simple view of class abundance. The trained local classifiers, combined with the conditional probability tables learned from the data, provide a complete representation of the hidden node Bayesian Network, on which inference can be performed.

The ability to interpret local classifier performance and class abundances from the conditional probability tables permits the following kinds of interpretation to be
trivially performed, despite the complexity of having 148 hierarchically structured
classes. Table 6.2 demonstrates that while the Biota texture node performs quite
well, the Sponges classifier does not. The fact that essentially no instances had a
predicted probability greater than 50% is a result of applying the calibration step on
unbalanced data. In order to have a group with e.g. predictions at 90% probability,
the classifier would need to be incredibly discriminative (given only 5% of the Biota
class are Sponges), as illustrated in Figure 5.3. The values in the Sponges texture
conditional probability table show that there is some difference in classifier output
proportions when conditioned on either presence or absence of Sponges, but far less
than for Biota texture. The effect of this is that inference will place less trust in the
Sponges texture classifier than the Biota one. Essentially, when the hidden node
Bayesian Network is trained on a calibration data set, it learns the accuracy of the
local classifiers on that data, and compensates; this built-in robustness to poorly
performing classifiers is a key feature of the hidden node approach, given hierarchical
classification will inevitably have poorly performing classifiers at deeper levels of the
hierarchy (with scarcer training data, and more subtle class differences).

6.4.4 Comparing Performance

6.4.4.1 Hierarchical Performance

Table 6.3 and Figure 6.5 show the results for each of the network configurations, with
the benchmark of the no evidence class prior predictions for a hidden node classifier.
The no-evidence case represents the accuracy of the classifier if the distribution of
classes is known accurately in advance (in this case, by fitting the conditional prob-
ability tables using the calibration set, which consisted of images drawn randomly
from the same data as the testing set).
6.4 Results and Discussion

(a) *Biota* hidden node

<table>
<thead>
<tr>
<th>id_2</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.75</td>
</tr>
</tbody>
</table>

(b) *Sponges* hidden node

<table>
<thead>
<tr>
<th>id_13</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>0.95</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.05</td>
</tr>
</tbody>
</table>

(c) *Biota* texture classifier node

<table>
<thead>
<tr>
<th>txt_2</th>
<th>0 (not biota)</th>
<th>1 (biota)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-25%</td>
<td>0.47</td>
<td>0.02</td>
</tr>
<tr>
<td>25-50%</td>
<td>0.12</td>
<td>0.02</td>
</tr>
<tr>
<td>50-75%</td>
<td>0.13</td>
<td>0.09</td>
</tr>
<tr>
<td>75-100%</td>
<td>0.29</td>
<td>0.87</td>
</tr>
</tbody>
</table>

(d) *Sponges* texture classifier node

<table>
<thead>
<tr>
<th>txt_13</th>
<th>0 (not sponge)</th>
<th>1 (sponges)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-25%</td>
<td>0.94</td>
<td>0.84</td>
</tr>
<tr>
<td>25-50%</td>
<td>0.06</td>
<td>0.15</td>
</tr>
<tr>
<td>50-75%</td>
<td>2E-5</td>
<td>5E-4</td>
</tr>
<tr>
<td>75-100%</td>
<td>2E-5</td>
<td>5E-4</td>
</tr>
</tbody>
</table>

**Table 6.2:** Example conditional probability tables for the 4-bin, calibrated, *siblings* hidden-node classifier. *id_* *denotes* a hidden/true node, and *txt_* *denotes* the texture based local classifier node. The columns represent the state of the parent node (0 or 1). The hidden node tables show that 75% of the data set was *Biota*, and 5% of the *Biota* was also *Sponges*. The classifier tables show that for e.g. the high performing *Biota* classifier predictions, 47% of the predictions fell in the 0-25% range when the example was not in fact *Biota*, and only 2% of the predictions fell in that range when it was. The *Sponges* classifier performs more poorly, with no predictions greater than 50% (other than the dummy predictions acting as a prior), and fairly minor differences when the true *Sponge* class membership changes between 0 and 1.

The scores in the table and figure can be compared directly against those in Figure 5.11.

**Key findings are:**

- Some of the uncalibrated approaches performed worse than the hidden node network that relied solely on class distributions (with no evidence from local classifiers).

- The *siblings* hidden node networks performed better than the discretised implicit node networks (whereas the *inclusive* hidden node network performed similarly). The implicit networks using raw probabilities performed markedly better.

- The abundance estimates on the first two layers of nodes (shown in Figure 6.6)
are in general very similar to the implicit Bayesian Network (and therefore superior to the heuristic approaches as described in ??). The results differ most from the implicit Bayesian Network on classes with broad bounds on the ground truth estimate, so it is unclear which of the Bayesian Network approaches are superior.

Figure 6.5: Hierarchical performance results
Table 6.3: Hierarchical performance results. For the combined Texture & Sensor network, All indicates that all sensor classifiers were used. Lvl 1 indicates that all texture, but only level 1 (Biota and Physical) sensor classifier outputs were used, which reduces the risk of overfitting the sensor classifiers on sparse depth data for rarer classes.

### 6.4.4.2 Abundance Estimates

The abundance estimate for the hidden node Bayesian Network was very similar to the equivalent implicit Bayesian Network result, as shown in Table 6.4 and Figure 6.6. The excellent results for the no observation case where no classifier outputs were used are unsurprising. In this case, the network falls back to estimating the prior class abundance from the calibration set for every point. The result is the poor discriminative performance in Table 6.3 and Figure 6.5, and near perfect estimate of overall abundance.
Figure 6.6: Abundance estimates for classes in the first three layers of the tree, as per Figure 5.12, with hidden node approaches included. The NoObs approach corresponds to not providing any local classifier outputs. The Texture approach is the best performing texture discretisation, described in Table 6.3.

6.4.5 Regional Tuning

Table 6.5 shows the same example conditional probability tables as Table 6.2, tuned on the Western Australia data only. Compared to the nationwide data set, the Western Australia tuned network:

- Contains much more *Biota* (87%, rather than 75%)

- Has a much smaller fraction of *Biota* also tagged as *Sponges* (2%, rather than 5%)
6.4 Results and Discussion

| Training | Strategy     | Calibrated   | Mean $|\epsilon|$ | N Underestimated | N Overestimated |
|----------|--------------|--------------|-------------|-----------------|-----------------|
| Siblings | BN           | Uncalibrated | 0.00015     | 13              | 3               |
| Siblings | BN           | Calibrated   | 0.00033     | 19              | 3               |
| Siblings | HBN.NoObs    | Calibrated   | 0.0002930   | 15              | 4               |
| Siblings | HBN.Texture  | Calibrated   | 0.0003313   | 52              | 3               |

Table 6.4: Results for the hidden node Bayesian networks, with no observations (HBN.NoObs), and texture classifier observations (HBN.Texture). Columns have the same meaning as Table 5.3, with the results for Siblings Bayesian Network repeated for easy comparison.

- Had poorer performance on the Biota local classifier
- Had negligible discriminative power with the Sponges local classifier

Due to the manner in which the calibration and testing sets were created (random selection of images from the BENTHOZ-2015 public data set), the conditional probability tables are a fairly accurate representation of the actual distribution of classes and classifier performance. This represents the best case; the other methods of obtaining the tables (such as expert guesses) would be less representative of the final abundances.

The class abundances from the id_* tables permit more accurate abundance estimates over the hierarchy. The classifier performance in the txt_* tables permits the network to effectively ignore classifier outputs that are known to perform poorly in a given region (such as Sponges texture in WA), or boost the importance of classifiers that perform particularly well.

The results on the regionally tuned Bayesian Network were greatly superior to the untuned ones, and even attained comparable performance to the implicit networks that used raw probabilities. If the performance decrease caused by discretising the implicit Bayesian Networks is at all suggestive of what occurs with hidden node networks, it would be reasonable to suggest that a regionally tuned hidden node network with better discretisation (or continuous nodes) would have superior per-
Chapter 6  Hidden Node Bayesian Networks for Hierarchical Classification

(a) *Biota* hidden node  (b) *Sponges* hidden node

<table>
<thead>
<tr>
<th>id _2</th>
<th>0</th>
<th>1</th>
<th>id _13</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>0.13</td>
<td>0</td>
<td>1.0</td>
<td>0.98</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.87</td>
<td>1</td>
<td>0.0</td>
<td>0.02</td>
</tr>
</tbody>
</table>

(c) *Biota* texture classifier  (d) *Sponges* texture classifier

<table>
<thead>
<tr>
<th>txt _2</th>
<th>0-25%</th>
<th>5E-3</th>
<th>25-50%</th>
<th>8E-3</th>
<th>50-75%</th>
<th>0.05</th>
<th>75-100%</th>
<th>0.94</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-25%</td>
<td>0.17</td>
<td>5E-3</td>
<td>0-25%</td>
<td>1</td>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-50%</td>
<td>0.09</td>
<td>8E-3</td>
<td>25-50%</td>
<td>3E-4</td>
<td>2E-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50-75%</td>
<td>0.14</td>
<td>0.05</td>
<td>50-75%</td>
<td>3E-5</td>
<td>2E-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75-100%</td>
<td>0.60</td>
<td>0.94</td>
<td>75-100%</td>
<td>3E-5</td>
<td>2E-3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.5: Conditional probability tables (as per Table 6.2), tuned for regional performance on the Western Australia subset of the calibration set.

formance to all the implicit approaches. This is a promising result for easily adapting a generic set of local classifiers (that have perhaps been trained on a large data set), for superior performance in a specific application such as an AUV campaign.

### 6.4.6 Sensor Redundancy

The combined sensor and texture hidden node network with 3-bin discretisation for both local classifier types was found to produce the best score on the test set. Table 6.6 shows the results after inference was performed on the same test set used in previous experiments. The local classifier predictions were provided as evidence for a varied proportion of the instances, to explore the behaviour of the network on sensor failure.

As expected, the results for the combined hidden node network are the same as either the sensor or texture only network, if only that evidence is supplied. The sensor classifier improves substantially over the no-evidence case, indicating that it is worth using as a backup (although in this case the texture only classifier per-
forms more effectively without the depth classifier present, indicating it introduces more noise than additional predictive power above the information in the texture classifiers). Also as expected, the simulation with each set of classifiers working half the time (random subset of data points with either texture or sensor classifier working) produced an intermediate result approximately halfway between the sensor only/texture only results.

One explanation for the lack of improvement with the additional classifier modality is that the sensor classifier was based on a single variable (depth) with extensive smoothing applied to reduce the potential for overfitting. With a more predictive modality, it should prove advantageous to have evidence from both classifier types operating simultaneously. Even so, the use of both modalities does provide some redundancy above the fallback to the “no classifier” case. Lastly, the hidden node Bayesian Network is the only one which does have a natural fallback to a reasonable estimate in the case of total classifier failure, whereas the implicit Bayesian Network, greedy walk, and others are simply unable to compute a prediction of any kind.

<table>
<thead>
<tr>
<th>Evidence Supplied</th>
<th>Brier</th>
</tr>
</thead>
<tbody>
<tr>
<td>No classifiers</td>
<td>0.01522</td>
</tr>
<tr>
<td>Sensor classifier only</td>
<td>0.01366</td>
</tr>
<tr>
<td>Texture classifier only</td>
<td>0.01056</td>
</tr>
<tr>
<td>Random 50% sensor only, 50% texture only</td>
<td>0.01210</td>
</tr>
<tr>
<td>Full sensor and texture outputs</td>
<td>0.01118</td>
</tr>
</tbody>
</table>

Table 6.6: Sensor redundancy test results

6.5 Conclusion

In this chapter, we moved beyond the implicit Bayesian Networks described in Chapter 5, to a more complex Bayesian Network structure that has a number of advantages. Performance was compared against other hierarchical classification tech-
niques (most notably the implicit networks in Chapter 5), and a number of different configurations were tested. The hidden node performance was superior to a similarly discretised implicit Bayesian Network. Using a manually selected discretisation, or continuous probability distributions (e.g. Gaussian) for the classifier nodes may improve performance beyond that of the implicit networks without discretisation. This would, however, come at the cost of reduced interpretability of the model. Model interpretability can be important for scientific applications (where understanding process and biases is vital), or for fieldwork where rapid troubleshooting by non-experts may be required. The properties of sensor redundancy, regional tuning, easy interpretation of class probabilities and classifier performances, and ability to include new sensor modalities without retraining make this an attractive framework for further work in classifying scientific taxonomies using an AUV.
7 Conclusion

7.1 Summary of Contributions

7.1.1 Hierarchical classification of benthic imagery

In Chapter 2, automated species detection from AUV images using a single class was explored, and a range of image features and classifiers was tested empirically. The key novel aspects were in the scope of the analysis, looking at generalisation ability to different AUV deployments on a large data set, rather than just a one-shot classifier built on a single deployment. In terms of classification, it was found that a moderately large patch (of order $100 \times 100$ pixels) using local binary patterns typically gave good performance. The finding was that training on a single deployment (as is often done) typically gave poor generalisation results. On the other hand, when using a “leave-one-out cross-validation” style approach (holding out one deployment at a time for testing), it was found that the ability for a classifier to generalise to unseen deployments was promising (at least within the same campaign).

The extension in Chapter 3 was to make use of the scientific taxonomy defined by marine scientists in [2], and treat the identification of benthic image content as a hierarchical classification problem for the first time. A challenging problem was
identified, of utilising expert classifications that were frequently only specified to partial depth in the hierarchy (e.g. as *Biota* or *Macroalgae*, rather than *Ecklonia radiata*). A modification was made to the common hierarchical performance measures (not penalising or rewarding predictions in the descendant tree of the expert labelled class), and a variety of commonly used hierarchical classification frameworks were applied.

### 7.1.2 BENTHOZ-2015 Public Data Set

Having completed the background work of Chapter 2 and Chapter 3 using a large Tasmanian data set, the limitations of data set size, availability, consistency and diversity became clearly apparent. In order to facilitate future research, a collaboration was formed between five research institutions to publish a publicly available data set of AUV data and expert scientific annotations. The raw data was obtained from the four marine science groups, processed, and mapped to the recently defined CATAMI hierarchy [2]. With approximately 400,000 point labels from 148 classes on 10,000 images, and a range of diverse habitats around Australia’s coast, this data set has the potential to form the basis of a large body of research into the automated classification of benthic imagery [13].

### 7.1.3 Bayesian Networks for Hierarchical Classification

In exploring the hierarchical classification literature in Chapter 3, it was found that one of the most flexible and commonly used approaches involved training individual classifiers to predict membership of each class in the hierarchy. Solutions using this *local classifier per node* approach were, however, quite varied, typically relying on heuristic algorithms and empirical testing to make predictions from the local
classifiers that would conform to the definition of the hierarchy. The novel contribution was to apply Bayesian Network theory to explain and refine these heuristic approaches, resulting in a stronger theoretical justification, and improved empirical performance on the BENTHOZ-2015 data set. The prediction framework involves treating the class hierarchy as a Bayesian Network, and using an assumption that probabilistic classifier outputs represent the conditional probability of class membership given the parent. One novel aspect was the use of post hoc calibration techniques on the classifier outputs to meet this assumption. Calibration of classifier probabilities is seldom, if ever, discussed (or applied) in both the benthic and hierarchical classification literature. However it is of fundamental importance in obtaining accurate estimates of overall class abundance, which is one of the primary motivations for automated classification in benthic ecology. It is also necessary in order to make reasonable comparisons between classifier outputs in the more general hierarchical classification domain, and was found empirically to improve performance. While reasonable calibration may be achieved without correction by a classifier network trained on siblings data with particular algorithms (as noted in [69]), it is prudent to at least check the calibration via the decomposed Brier score, rather than assuming the training process satisfies the conditional probability assumption.

### 7.1.4 Hidden Node Bayesian Networks

In Chapter 6, the use of Bayesian Network theory was extended to a more complex network structure that separated a hidden “true” node from nodes representing local classifier predictions. Performance was compared against other hierarchical classification techniques (most notably the implicit networks in Chapter 5), and a number of different configurations were tested. Results suggest that superior
performance is possible with this approach, and the flexibility and other benefits make it well suited to forming the basis for classifying scientific taxonomies with AUVs.

7.2 Extensions, Applications and Future Work

The goal of this thesis was to investigate areas of research required to create an automated classifier for AUV data, that could be used either in situ for adaptive planning based on semantic understanding of the environment, or to reduce the manual effort required by marine scientists. A hierarchical classification approach has been defined that can be easily tuned for new deployment locations, deal with multimodal sensor input (in a way that is robust to intermittent failure), provide accurate estimates of class abundance for ecological applications, and conform to a precisely defined scientific taxonomy of classes defined by marine scientists. As with all such complex fields of research, a number of promising avenues could be investigated, outlined below.

7.2.1 Multimodal sensor inputs

The hidden node Bayesian Network approach described in Chapter 6 is ideally suited to including classifiers based on different sensor modalities, in a modular fashion. Some AUVs have a stereo camera pair, which means that 3D features (such as [45]) could be computed and included as input to a separate set of classifiers. There are many situations in which a 3D surface cannot be computed, so the network’s robustness to missing classifier inputs is important. Many AUVs also include acoustic backscatter sensors, and there is an emerging body of research looking at the utility of dedicated hyperspectral imaging sensors for colour correction and species
classification on board AUVs [18]. A network of local classifiers could be trained on each of these modalities, with observations included based on either presence of the sensor on the AUV, or (in the case of stereo 3D features), whether the classifier can provide a valid output at a given point in time.

### 7.2.2 Enhanced local classifier performance

A deliberate decision was taken to place as little emphasis as possible on the improvement of local classifier performance, due to the concurrent developments in the field of deep learning (resulting in either wasted effort in carefully refining traditional computer vision techniques, or the unmanageable task of keeping pace with the rapid progress of deep learning techniques, while making contributions to hierarchical classification).

As such, the majority of this thesis relied on a network of classifiers, each trained in the same way (using greyscale local binary patterns as descriptors of a $127 \times 127$ pixel patch). This provided a consistent baseline against which to benchmark hierarchical classification techniques, on a set of classifiers with performance ranging from moderately good (at the Biota and Physical level), to those that performed barely better than random selection at lower levels of the class hierarchy. It would be highly worthwhile to make use of recent major developments in the field of deep learning to either construct features for individual classifiers, or train classifiers directly. With recent results in the computer vision literature, and machine learning competition arenas such as Kaggle, it would be reasonable to anticipate substantial improvements in base classifier performance.

Another enhancement could be to create a superpixel model (as in [43]) by oversegmenting the images such that areas of similar appearance (such as sand, or parts of benthic organisms) are separated from each other by a segmentation algorithm.
This could permit better abundance estimates from small samples than the relatively large square patches used throughout this thesis.

Expert knowledge could be beneficial when used in tandem with these automated feature learning techniques. One example might be to use a hyperspectral image sensor to differentiate between two rare classes based on expert knowledge of their absorption spectra, rather than relying on large amounts of labelled data to be produced.

### 7.2.3 Adaptive Planning

One key drawback of using an AUV over human dive surveys is semantic understanding. A scientific diver can quickly see whether the selected site meets the scientific goals. If the goal is to “explore a kelp forest known to be roughly in this area”, it is entirely possible for the AUV to capture an entire mission of sand adjacent to the forest, due to the lack of semantic feedback. The waste of actual mission time is compounded with the time required to set up the mission, reaching the sea floor, recover the vehicle, charge and reset ready for redeployment.

Potential applications of onboard classification include:

- Including semantic goals in an AUV mission plan, such that the AUV can automatically abort the mission if they are not met (e.g. define a geographic path as normal, and note that the dive is intended to be on a kelp forest).
- Using an acoustic modem to provide feedback on dive progress. While the bandwidth of an acoustic modem is too low to transmit images back to the ship during a survey, it would be possible to transmit summary statistics for semantic content in the dive as it progresses. This would allow researchers to abort a mission manually if desired, or even plan and deploy multiple AUVs
in succession, as the area is explored.

- Adaptation of the path plan based on semantic content. In an unexplored area, it is common to instruct the AUV to survey in a straight line for several kilometres (known as a “transect”). Subsequent dives may then be planned to cover areas of interest in more detail. With onboard classification, the AUV could autonomously add this extra coverage while doing the transect (some relevant work mapping hydrothermal vents was performed in [53]).

- New types of purely semantic missions could be defined, such as following the boundary between sand and kelp forest.

Given the expense of running the AUV from a research vessel, onboard classification has the potential to significantly reduce the cost of running AUV surveys by making individual missions more effective, and greatly reducing the risk of wasted missions that capture irrelevant information (such as large tracts of sand).

The term “adaptive planning” is used in two main contexts with AUVs. *Inter-mission planning* (such as in [74, 11]) seeks to find the optimal location and plan for a new mission, based on some general model of the habitat. This model may be derived from classification on previous dive images, ship-borne multibeam surveys over the area, satellite data, or other means.

Adaptive planning in this work refers to *intra-mission planning*, where the aim is to adapt a dive mission on-the-fly in response to the data stream being collected by the AUV.

In [11], the benefit of an AUV adapting a mission in situ is studied, given an unknown exploration environment, and limited communication ability with the operators. A method for following the boundary between two habitats is proposed, and tested on simulated data (with the assumption of a sensor that can accurately differentiate
between habitat types). In [94], a proposal and field tests are documented, for a strategy that reacts to sidescan sonar data being collected by an AUV, aimed at maximising the quality of the data, rather than the semantic content. In [72], an adaptive sampling method for an AUV testing water column data is designed and field tested.

Very little work has been done in the field of adaptive planning based on real time classification. In [38], the authors note that they were unable to find any literature reporting real-time habitat classification in AUVs. The literature in the adaptive planning/sampling field typically either performs simulations assuming an accurate habitat classifier [11], or extends to field tests with very simple sensor driven tasks, such as following thermal gradients [72, 93].

The most recent relevant work is documented in [46], which builds a semantic perception model of the environment using topic modelling, and then plans a path to maximise semantic information acquisition. This was tested in the field using *Aqua*, an amphibious AUV to follow a diver, and conduct sea floor exploration. The scarcity of field tested algorithms among benthic following AUVs is perhaps due to the complexity of detecting benthic habitats using imagery (in both a computational and a model accuracy sense).

Earlier, field trials were conducted with the Starbug *AUV* in [38], using whole image classification (using features extracted from automatically detected keypoints). The classifier made a limited hierarchical decision, first choosing between (fish, sea grass) and (sand, rocks), and then choosing the sub-category. Performance is reported as around 80-90% accuracy at the whole image level, on the training set of 800 images. They then implement the algorithm in real-time on the *Starbug AUV*, computing features on 96 keypoints in each image (at a computational cost of 0.64s per image). The algorithm is then tested in a purely observational sense - the
real-time classification has no impact on the mission plan of the AUV.

### 7.2.4 Semantic and Visual Similarity

One question raised by [15] is the potential for mismatch between the semantic hierarchy defined by the intended uses of the classifier, and the actual visual similarity of the classes. On synthetic data, they find that a local classifier per node approach benefits most when the semantic hierarchy aligns well with visual similarity. Both Chapter 5 and Chapter 6 suffer to some extent from a scientific taxonomy that frequently differs substantially from visual similarity (e.g. screw-shells look very much like coarse sand, however they are located in the Biota: Molluscs class). A potentially interesting line of investigation would be to permit the Bayesian Network (or even an undirected graphical model) to learn a structure from the data that better represents the visual similarity, and may perform better. This would then need to be reconciled after prediction with the need to predict according to the scientific taxonomy. Given the somewhat arbitrary structure of the scientific taxonomy (where decisions need to be made to order by morphology, then colour, then species, etc.), this process could be used to suggest a more suitable class structure, where the classes still meet appropriate scientific definitions.

### 7.2.5 Biodiversity Metrics

One interest in the field of benthic ecology is in biodiversity metrics; measures of how many different species, or types of organisms, exist within a habitat (higher biodiversity is often used as a proxy for good health of an ecosystem). A biodiversity measure based on hierarchical classification would not be required to accurately pinpoint the precise locations of individual species; rather, it would need to identify how
many are present in an area. Given the hierarchical classification scheme described in this thesis tends to predict abundances well, but suffer from poor predictive accuracy for the more specific classes, it may be a promising basis for constructing such a metric.

### 7.2.6 Changing hierarchy over time

In the pursuit of a truly reusable classifier for AUV data, it is important to recognise that the structure of the semantic hierarchy may change over time. New nodes may be added to recognise new groups or species, nodes may be removed, merged or rearranged. Notionally, the hidden node hierarchical classifier is most suited to these adaptations, as it is an almost trivial exercise to recompute the conditional probability tables using the classifier outputs, learning the relevance (performance) of the local classifiers to the new classes. New individual classifiers could be created and added incrementally, without the need to rebuild them all from scratch.

Similarly, the conditional probability tables can be used as a measure of when classifiers begin to perform poorly (perhaps due to changing environmental conditions, or degrading hardware), such that optimisation can be performed for individual classes.

The most efficient way to modify the hierarchy and replace individual classifiers would be a valuable area of research, once the basic classification task is performed sufficiently well to be used in practical applications.
Bibliography


[88] Su, C., and Hsu, J. An extended chi2 algorithm for discretization of real value attributes. IEEE Transactions on Knowledge and Data Engineering 17, 3 (2005), 437–441.


modified hierarchical Brier score The Brier score, as defined in 5.3.4.1, measures the mean squared difference between the predicted probability, and binary class value. The hierarchical modification relies on the same definition of true positives, false positives and false negatives as the modified hierarchical f1-score.

modified hierarchical f1-score The f1-score is the harmonic mean of precision and recall. The modified hierarchical f1-score makes use of the hierarchical definition of true positives, false positives and false negatives, defined in detail in Section 3.2.4.