Large Scale Multi-output and Probabilistic Benthic Habitat Mapping

JUSTIN TING

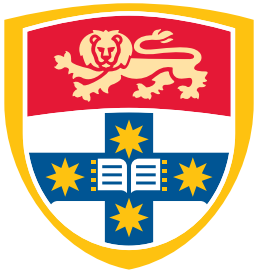
SID: 430203826

Supervisor: Dr. Simon O'Callaghan

This thesis is submitted in partial fulfillment of
the requirements for the degree of
Bachelor of Information Technology (Honours)

School of Information Technologies
The University of Sydney
Australia

31 October 2016



THE UNIVERSITY OF SYDNEY

Student Plagiarism: Compliance Statement

I certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);

This Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: Justin Ting

Signature: **Date**:

Abstract

Being able to predict the state of benthic habitats based on limited information is crucial for environmental conservation, particularly as the impact of human activity on our oceans is greater than ever before. Prior to the flourishing state of machine learning we see today, creating benthic habitat maps would have been an opinionated and subjective task carried out by experts, based on limited data sampled from habitats. In more recent years, the number of tools which allow convenient application of machine learning to data sets has seen more research carried out in the area of benthic habitat mapping. Many of these such as Support Vector Machines (SVMs) are deterministic, in that predictions at a location, given some input data, strictly provide a single label as its guess, without any information regarding how certain it is. Probabilistic models such as Gaussian processes overcome this and provide predictive variances with predictive means, adding a rich layer of information that represents the confidence of the predictions made. But due to the mathematical steps involved, the time it requires to fit Gaussian processes exceeds practicality beyond several thousand points. To allow the use of this Gaussian processes to scale with large data sets, this study explores the viability of approximation methods in creating habitat maps. The methods mentioned so far only account for bathymetry points corresponding to a single label, despite working with multi-label data, meaning not all the information available is being fully utilised. Dirichlet Multinomial distributions are able to model these multi-label outputs, and the method that is used to exploit the full extent of the class data used in this study.

Acknowledgements

refer to meeting notes to include acks for all the existing work used for this thesis

Contents

maybe put a notation section?

CHAPTER 1

Introduction

Earth's oceans cover 70% of its surface, but only less than 10% of the Earth's oceans have been explored to date[1]. There have been increasing efforts over the past few decades to more efficiently map out these unexplored areas to monitor marine ecosystems to be able to track the state of them over time for reasons such as management and preseveration purposes. The process used is called benthic[2] habitat mapping and involves generating predictive maps of different habitat types at the ocean floor. Most efforts to create benthic habitat maps share some basic key steps - acoustic data is used to estimate properties about the surface of the benthos, then its relationship with other 'truthing' data such as images or video are inferred.

Early attempts at this understanding and modeling these relationships would have been expert-driven where any map created by an individual or individuals would be fully subject to their personal biases and knowledge. With the lack of better tools available, this was the best option - until machine learning algorithms began to be formulated more formally, and applied to real world problems. Thus, deterministic methods such as random forests and support vector machines made their debut, and enabled predictions on large sets of potentially complex data that would have previously proved to be a considerable challenge or even impossible when done manually, and would have required major compromises.

However, considering that the creation of habitat maps would be for conservation of marine management purposes, it would be important to be able to state the confidence of the predictions are made, a property that deterministic methods do not possess. Gaussian processes are a

[1] Oceanservice.noaa.gov. (2016). How much of the ocean have we explored?. [online] Available at: http://oceanservice.noaa.gov/facts/exploration.html

[2] Benthic refers to things relating to the bottom of the ocean, i.e. the ocean floor

complex model that can provide this information - providing both the probability of every label being the 'correct' one at a certain data point, as well as the variance on each of these predictions. Due to the $O(n^3)$ matrix inversion steps involved in them, the time that would be required to fit a Gaussian process to more beyond several thousand points would be astronomical and impractical for any real world applications. To account for this, there exist approximations such as product of experts and Bayesian committee machines that allow levels of parallelism of the model (that are inapplicable to the basic Gaussian process) that are (almost) bounded only by hardware.

These methods so far ignore the fact that asynchronously collected bathymetry and image data will often not correspond to one another in a $1:1$ manner (as is the case in this study), with each bathymetry point likely related to several images. This results in a dataset where for each coordinate with bathymetry data, the labels across several labels are attributed to it, or in other words, a vector of label counts. As the aforementioned models do not handle such data, approximations were made where these vectors were simplified down to a single label, throwing out a majority of the information that was collected. To actually fully utiliese these vector counts, a method is needed that can model this data appropriately - such as the Dirichlet Multinomial. This allows predictions to be made over category counts, providing normalised predictions at each point, not only correctly interpreting the data in contrast to other methods used in benthic habitat mapping, but also allowing properties of the benthos such as biodiversity and co-habitation to be directly inferred from the data, not requiring additional post-processing steps.

1.1 Contribution

The main contributions of this thesis are twofold. Fistly, an investigation into current probabilistic methods is performed. Gaussian proceses are a state-of-the-art method in this regard, although the implementations of them available in open source libraries and the like are currently unable to work with datasets larger than several thousand points without hitting computational limits, which makes them impractical for real-world use, if, for example, trying to fit

a model for data prediction alone for several tens of thousands of points could take weeks, if not longer. This work seeks to investigate whether recent innovations in the field are able to improve performance sufficiently that it can be used on larger datasets, and in particular, to the specific field of benthic habitat mapping.

The second aim is to explore the use and modelling of information where input data does not only have a single label such as 'sand', but could have a category count over labels, such as 7 counts of sand, 3 counts of coral, and 1 count of rhodliths, also representible as a normalised distribution $[0.7, 0.3, 0.1]$. Such data is collected as a result of automated underwater vehicles (AUV) collecting images at the ocean floor at a density of, for example, one every $10m^2$, whereas bathymetry data is collected from the ocean surface at a lower rate, such as $50m^2$. This means there can be more than one image, and hence habitat label per bathymetry data point, where any single image is 'assigned' to the nearest bathymetry data point. While this can be used with single-output machine learning algorithms, it involves simplifying every data point's label distribution to only the most dominant label - in the aforementioned example, $[0.7, 0.3, 0.1]$ would be simplified to sand. Although this allows common and readily available methods to be used, it also discards a considerable amount of otherwise useful information. To fully utilise all this information, use of the Dirichlet Multinomial distribution to model the data is explored, along the other information that can be gathered from it such as measurements of biodiversity and entropy to represent confidence in predictions.

1.2 Motivation

The motivation for exploring the aforementioned technical aspects of benthic habitat mapping is to explore how the process of benthic habitat mapping can be performed more economically while also providing richer information than has generally been the case in such efforts, and also automating as much of the process as possible. To break down the three aims mentioned - the economic aspect comes into play in terms of density at which the data is collected. Naturally, collecting more data would cost more money - being able to work with coarser data where multiple images may correspond to single bathymetry data point using a technically

sound method that can make full use of the data has the potential to cut down on the need to collect huge amounts of data. Richer information can be obtained both from the probabilistic component of Gaussian processes as well as the Dirichlet distribution's entropy from the Dirichlet multinomial. This is necessary because any actions taken to manage or conserve benthic habitats inherently comes with a level of risk whether it be environmental or economical, and quantifying this risk would require that at the very least, the mathematical models used to form predictions need to be able to state their confidence.

1.3 Outline

We will first review some of the existing literature in **??**, on the collection of bathymetry and image data briefly, then on deterministic approaches to benthic habitat mapping to date, such as logistic regression, and random forests, and their performance on varying types of benthic environments. This is in contrast to the more informative probabilistic and multi-output approaches that will be explained in Chapter 3 and Chapter 4, where we look at the mathematical background behind Gaussian processes and Dirichlet multinomial regression. In Chapter 5, we then apply the techniques explained in the previous chapters and observe their performance, points of interest, as well as how the information obtained differs to methods visited in explored in the related works. This is followed by Chapter 6, an evaluation of the experiment designs, as well as the limitations that were present and how they could have affected the study. The study is then concluded and summarised in Chapter 7, discussing the possible impact of this work, as well as areas of the study that could be explored in further detail as possible future work.

CHAPTER 2

Related Work

In this chapter, an overview of will be provided of what benthic habitat mapping is and the general steps involved in data collection, followed by a review of some of most commonly used approaches when performing benthic habitat mapping. The practice of benthic habitat mapping precedes the rapid developments in machine learning methods in recent history, and as such, early attempts would naturally have involved manual predictions based on available information that would be subject to the biases of experts involved in the process. It is thus expected that given the same raw data, different experts who have had varying experiences in their field would come to different conclusions.

This phenomena was observed in a geoscience related study (Bond et al., 2007), where over 400 individuals with geoscience backgrounds were asked to assess a synthesised seismic image, with just under 25% correctly identifying the 'true' tectonic setting and the three main fault strands. Interestingly however, (inexperienced) students were as likely to give incorrect responses as those with over 15 years' experience, where the latter often drew conclusions linked to the area that the held expertise in. Early efforts to create benthic maps dating back to the at least the 1980's(Gibson et al., 2007) followed this trend, where the lack of more formalised approaches meant experts would use the available data to extrapolate habitat maps based on the understanding that they had. This points to the variability of expert-driven modeling of natural environments such as benthic habitats, and the need for data-driven techniques, where expert input can be used as a supporting source of information rather than the only, or dominant one.

2.1 Benthic Habitat Mapping Overview

The process of benthic habitat mapping involves three key steps that the large majority of all studies in the area go through(OzCoasts, 2015). In this section, a brief overview of each of these steps will be given, along with common procedures involved. Habitat characterisation extracts the physical properties of an environment, whereas classification uses the data available to classify or cluster the raw data into habitat groups. Habitat mapping would then involve the modelling of relationships between the existing data, and in the case where the availability of one source far exceeds another, the relationship determined at the overlapping areas is then extrapolated to where only one data source exists, to formulate a habitat map.

2.1.1 Habitat Characterisation

If high resolution, multimodal[1] data for the entire ocean's benthos was easily obtainable, creating benthic habitats maps for any given area would be not be an incredibly difficult task. As this is prohibitively expensive, the altnerative is instead collecting relatively large amounts of economically obtainable information such as bathymatery data, and comparatively fewer samples of data that are costly to collect such as images (so that a relationship can be modeled between them, to be explained below). This subsection provides a brief summary of data collected and methods used to do so.

As exhaustively exploring Earth's ocean floor with underwater vehicles to capture it visually as well as all its physical properties is an infeasible task, compromise is required so that modest amounts of data can be collected economically, whilst still being informative. Remote-sensing data is thus used, usually obtained from acoustic backscatter methods, involing the firing of sound waves towards the benthos, where their frequency and strength upon returning is used to deduce the depth of the area from where it rebounded (and in turn, allowing other properties to be inferred, such as slope and rugosity[2]). Shipborne tools such as single-beam echosounders (SBES) and multi-beam echo sounders (MBES) facilitate the collection of this data in the form

[1]Multimodal data refers to the different *information* that resides within it - e.g. an area of benthic terrain can be represented by a single photograph, or numerical representations of its physical properties.

[2]rugosity is the measurement of surface roughness

of acoustic backscatter, with the latter being the modern alternative that can collect larger quantities of higher quality data more efficiently(Calvert et al., 2015). Since bathymetry data alone is not enough to create habitat maps as this would incorrectly assume that a limited set of physical properties alone can fully explain a habitat(cite), truthing data also needs to be collected to verify habitats. Before underwater vehicles equipped with any number of capabilities required by researchers and the like were as readily available as they are today, collecting sediment samples would be a common method to verify habitats(cite). Given the current technology, images are easier to collect via use of autonomous underwater vehicles (AUVs) that can be more readily processed afterwards as well.

However, since AUVs operate at the benthos, and methods such as MBES for collecting bathymetry data occurs at the ocean surface, it would be expected that the exact coordinates at which they are taken would not match up exactly. To account for this, in the case where multiple images exist around any given bathymetry point, all these images are then attributed to it, as in Figure 2.1

2.1.2 Habitat Classification

Almost all studies use *in situ*[3] data to complement the acoustic data for building a model between these two sources. However, this data needs to be labelled with the habitat they belong to such as 'bedrock covered by discontinuous seagrass cover' or 'Maerl interspersed with gravel' (Micallef et al., 2012), with the algorithms for doing so falling under either the supervised or unsupervised categories.

Studies have used both unsupervised clustering(cite) and supervised clustering (classification)(cite) to label the truthing data for the model-fitting stage. Often, there may be large amounts of visual data, beyond that which any human or even team can reasonably, manually cluster - and as such, unsupervised algorithms are first used to create these clusters, after which an expert may be brought in to verify/simplify (or otherwise) the resulting clusters. One possible method

[3]in situ, in a biological context, refers to the precise spot in which something occurs. In a habitat mapping context specifically when referring to truthing data, it simply means, in the case of bathymetry and image data, images taken at the exact spot corresponding to a particular bathymetry sampling location.

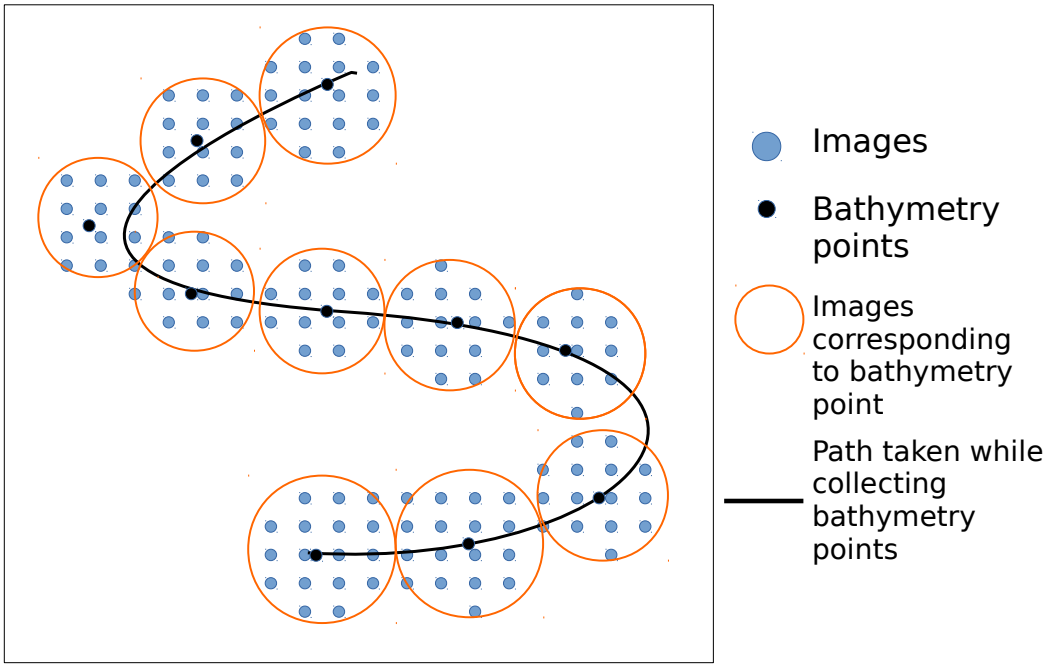


FIGURE 2.1. An example of bathymetry data and image collection. As the density of images are higher than the bathymetry data, the method of utilising both sources fully is that every image is assigned to the closest bathymetry point (within reason), resulting in each of them taking on a number of habitat labels, depending on how the labels vary within a given area (the orange circle, in this case).

in the unsupervised category is to use hierarchical clustering as seen in Pizarro et al. (2009). This is an approach in which a layered tree[4] is formed where the two nearest clusters (based on a pre-deinfed distance metric) are grouped together to form a larger cluster, with a base case where every point is its own cluster. The distance metric used in this particular study was the Kullback-Leibler convergence between points, where each was attributed to a distribution of a set of features, including properties such as saliency that was calculated using colour and texture of the image, as well as colour histograms of comprehensively normalised images.

[4]the tree described here is formally known as a *dendrogram*

Through more complex approaches, the supervised and unsupervised paradigms can be combined to get benefit from the advantages of both - reducing manual human input required, whilst also directly incorprating a human's domain expertise. The Dirichlet Variational Processes used in (Friedman et al., 2011) as a part of their 'active learning' is an example of this. By employing a probabilistic model over the image features during the unsupervised clustering process, every label is given a degree of un/certainty. The clustering algorithm can then be required to ask for a manual classification for a particular image if the level of certainty is too low or unsatisfactory which is fixed such that the model can no longer further modify this particular label.

2.1.3 Map Creation

The final step is map creation, where the labelled truthing data is combined with the bathymetry data to generate predictive maps corresponding to the area over which bathymetry data exists. This is the focus of the next section of this literature review, and of this overall study as well. There are two main ways in which acoustic bathymetry data collected can be used for map generation, as described in Ahsan et al. (2011). The first approach involves the direct clustering of the acoustic data, then retroactively collecting truthing data in the relevant locations to determine what physical habitats the clusters represented. This is inherently flawed, as it assumes that all the bathymetry information is close for the same habitat and far between habitats. A simple instance of this would be for two areas with sand at significantly different depths (with traces of other habitats too), potentially causing such an approach to identify one as sand, and the other as the trace habitat, if insufficient truthing data is collected.

The other approach is to first collect and cluster the truthing data before modeling its relationship with the acoustic data, where we apply this relationship to the areas without truthing data to create the habitat map. The latter approach is the one taken in this study, and the basis upon which the following review of techniques used for benthic habitat methods is based upon.

(example of a simple mapping effort on a synthesised dataset here - not too sure what to include independent of the next image though)

While the majority of modern papers in benthic habitat mapping employ machine learning techniques for map creation, this doesn't exclude those that do not from providing useful information and insight. An important study was undertaken in 2001 that employs relatively basic statistical analysis, employing different forms of variance as its main tool of analysis. (Kostylev et al., 2001), at the time (and in fact, even now) integrated more sources of data together than most other studies undertaken - multibeam bathymetric data, geoscientific data, seafloor photographs, habitat complexity, and relative current strength. Rather than using a single model to fit and train data as is more traditionally done in machine learning, multiple statistical tools are used in a peacemeal manner including One-way analysis of variance (ANOVA), Student-Newman-Keuls (SNK) tests, and Analysis of Covariance (ANCOVA).

Although little is done to address and verify accuracy of the actual results/map in this paper, it provides value through the analysis of variance and covariance performed on and between different benthic/marine properties, establishing relationships typically taken for granted or ignored. For example, it is established that while sediment type contributed heavily to a higher taxonomic[5] group count, there was little relationship between sediment type and depth. However, this only indicates that there is no linear relationship between the two, and doesn't necessarily preclude a non-linear relationship between them, perhaps with the inclusion of other parameters as well. In particular, Kostylev establishes that gravel subrates are more abundant with varying taxonomic groups than their sand counterparts.

Such findings provide useful insights for future studies that will allow a better assessment of data, or to be able to apply initial assumptions in obtaining better results. However, constantly seeking a deeper understanding through a proportionally increasing amount of sampling creeps towards 'exploring' the entire Earth's oceans manually. To obtain economically feasible yet reliable predictions from the limited data that we have, we need to formally employ machine learning techniques to further utilise the information that we gather.

[5]Taxonomy in a biological context is the categorisation of different organisms based on shared characteristics.

2.2 Deterministic Machine Learning Algorithms in Benthic Habitat Mapping

As benthic habitat mapping covers a diverse range of discplines, namely "marine biology, ecology, geology, hydrography, oceanography and geophysics" (Brown et al., 2011) in addition to statistics and machine learning, it is logical that it would take considerable effort and vast resources to give each relevant discpline an equal, and large amount of attention within any single study. Thus, different studies can rely on the collective findings of others to launch their own research and look further into particular lines of inquiry, start new ones altogether, or evaluate effectiveness of methods used in the field/etc. Within benthic habitat mapping, one prevalent line of inquiry is how to create more accurate, higher quality maps by employing machine learning techniques. For the remainder of this review, we will be looking at common deterministic machine learning techniques and their performance when used to generate habitat maps. A deterministic algorithm gives a single output or outcome given some input, while a probabilistic one will give a possible *distribution* of outputs and a measure of the likelihood of occurrence over this distribution.

Some of the most commonly used classification algorithms in benthic habitat mapping are also commonly used across domains involving classifying data in general. These include support vector machines (SVMs), random forests, k-Nearest neighbour classifiers, and logistic regression. Although these are more traiditional, more well known methods, they are still used in new research, for example in Wahidin et al. (2015), where all of the aforementioned algorithms were tested. Although empirical studies have been performed on these methods to compare their performance across many domains such as in Caruana and Niculescu-Mizil (2006), their effectiveness in benthic habitat mapping varies from study to study, depending on the benthic location that can affect variables such as the sort of habitats present and hence the overall physical properties a particular benthic region, how quickly habitats transition from one to another, or the level of biodiversity/mixing of habitats.

Multinomial logistic regression builds upon linear regression, a staple of machine learning algorithms. Its basic formulation only allows two classes to be predicted at a time - multiple

logistic regressors are required to train one class against the rest to generalise beyond only 2 classes. Though it was ranked towards the bottom of Caruana's average performance rankings, it has had some limited success in less complex binary problems where it was able to distinguish between hard-rock bottoms at the benthos with those that were not with 70% accuracy (Dunn and Halpin, 2009). In other instances, however, it was only able to get as low as 23% above a random guess(**?**), suggesting its unsuitability with the data that was available.

As the overall 2-nd ranked machine learning algorithm in Caruana's empircal study across 11 different problems, it is no surprise that Random Forests appears in many studies (Lucieera et al. (2013), Seiler et al. (2012), Hasan et al. (2014)), providing good results in contrast to logistic regression whilst being low in complexity. However, Lucieera et al. (2013) found that while random forest classifiers were most able to classify substratum and rugosity, K-nearest neighbour classifiers most accurately classified sponge structure classes, pointing to the need to do a more systematic comparison of different methods in benthic habitat mapping. This highlights the fact that even models that are considered state of the art can be suboptimal without necessarily being subject to extreme conditions. A further advantage to using random forests as pointed out in (Hasan et al., 2014) is that it can provide insight into which features were more important than others, which can aid future studies to be more successful and efficient by focusing more efforts towards collecting the most influential data. The success met with using random forests make it a good benchmark to compare against for future work that aim to develop methods to create more accurate benthic habitat maps than has been done before.

(need to add to this section of lit review)

2.3 Summary

In this chapter, we briefly looked at how the data used in benthic habitat mapping is collected, followed by a review of deterministic methods that have been used in existing studies. It is evident that due to the varying and even unknown nature of the intricacies of the benthos, there is not any single stand-out method that can be labelled as the 'best' option when choosing how to create predictive habitat maps. Given that the purpose of these maps are to be able to

economically map out what large swaths of the ocean look like so that better decisions can be made to conserve and protect them, the need for probabilistic mapping methods become apparent. Such methods are still able to generate labels for input data as we have seen so far, but also provides a level confidence per prediction, meaning that a statement such as 'this area of benthos is 75% sand and 25% coral', for example, could instead be more informative - 'this area is on average 83% likely to be 75% sand, and 74% to be 25% coral' (with the remaining probable distributions excluded for simplicity). Being able to estimate this uncertainty prior to taking any actions is essential to be able to calculate any inherent risk associated with it.

CHAPTER 3

Probabilistic Habitat Mapping

The methods of habitat mapping explored until now were mostly deterministic ones, where predictions were absolute, and as such did not provide a *level of confidence* in the predictions made, or in other words, probabilistic output. The partial exception to this was logistic regression, though the probabilities provided by it are absolute, with no variance to indicate confidence. Regardless, as a parametric method, the complexity of a logistic regressor must be defined beforehand, whereas a Gaussian process in simple terms allows the data to 'speak for itself'. More formally, this refers to a Gaussian process' non-parametric nature, meaning the data is incorporated directly into the model where new data can increase the confidence of the model.

In this chapter, we will look at Gaussian processes as a technique to generate predictive habitat maps. We begin by going over the basics of Gaussian process regression, and how a small extension/post-processing step extends it to allow Gaussian process classification. To train the hyperparameters of a Gaussian process, leave-one-out cross validation (LOO-CV) was used. They also need to define a *kernel* that governs the similarity between any two points, forming the full covariance matrix that described the relationship between all pairs of points. The kernel chosen was the squared exponential kernel, one of the most commonly used for Gaussian processes. Note that detailed proofs and derivations are not covered here, and interested readers should consult Rasmussen and William's Gaussian Processes for Machine Learning book (Rasmussen and Williams, 2006) for a definitive guide to all things Gaussian process related. In particular, Chapters 2 and 5 are of the most relevance, as they detail Gaussian process regression, and model selection and adaptation of hyperparameters respectively.

3.1 Gaussian Process Regression

Compared to deterministic methods like linear regression[1] that explains data by optimising $\mathbf{y} = \mathbf{X}\beta + \epsilon$, where $\mathbf{y}$ are the response variables, $\mathbf{X}$ are the input variables, and β are the regression coefficients, Gaussian process regression takes a Bayesian approach by adjusting probabilities when given more information (input data), and performs inference over functions.

We define a Gaussian process on input $\mathbf{x}$ to have a mean function m and covariance function k (or in other words, the kernel), where $\mathbf{x}$ and $\mathbf{x}'$ are the training and test inputs respectively:

$$f(x) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{3.1}$$

The chosen kernel is the squared exponential, represented by the covariance function between points p, q, where $\mathbf{x_p}, \mathbf{x_q}$ are the vector of features at each point is thus given by:

$$cov(f(\mathbf{x_p}), f(\mathbf{x_q})) = k(\mathbf{x_p}, \mathbf{x_q}) = \exp(-\frac{1}{2}|\mathbf{x_p} - \mathbf{x_q}|^2) \tag{3.2}$$

The free parameters involved in the squared exponential kernel are the length scales l, signal variance σ_f and noise variance σ_n, and need to be optimised to perform predictions on unseen data.

$$k(\mathbf{x_p}, \mathbf{x_q}) = \sigma_f^2 \exp(-\frac{1}{2l^2}(\mathbf{x_p} - \mathbf{x_q})^2) + \sigma_n^2 I \tag{3.3}$$

where σ_f is the variance in the training data, and σ_n is the variance of the Gaussian noise. The length scales l are not a single variable as the equation may suggest, but a vector of length scales equal in length to the number of dimensions in the inputs $\mathbf{x}$. If l was simply a vector of 1s, which would give every feature in the input space equal weighting, but it is not likely in a real world dataset for every feature to have equal importance. This is what the length scales account for - by tuning l_i for each feature i, the model can learn during the fitting process which features are important, and which ones are not.

[1] Strictly speaking, linear regression is a Gaussian process with a linear kernel - but due to its simplicity, and that Gaussian processes are not usually explained by weight coefficients but by their hyperparmeters, we are juxtaposing linear regression to Gaussian processes here as a simpler approach.

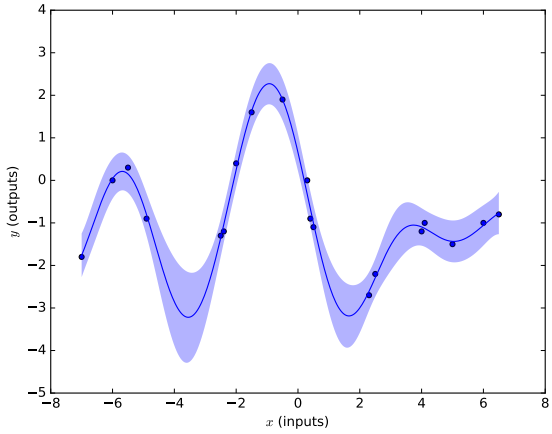
The equations for the predictive means and variances then incorporate the covariance function, and hence the hyperparameters, as follows:

$$\bar{f}_* = \mathbf{k}_*{}^T (K + \sigma_n^2)^{-1} \mathbf{y} \tag{3.4}$$

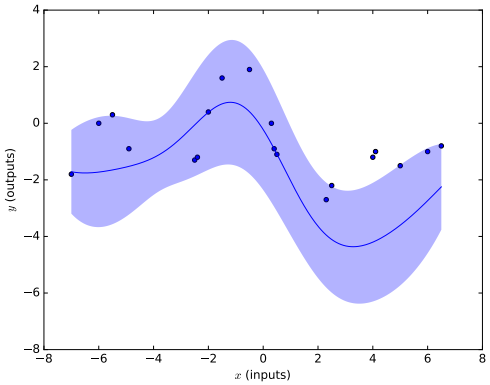
$$\mathbb{V}[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*{}^T (K + \sigma_n^2)^{-1} \mathbf{k}_* \tag{3.5}$$

where $K = K(X, X)$ is the covariance matrix over all training points, and $\mathbf{k}_* = \mathbf{k}(\mathbf{x}_*, X)$ is the covariance between a single test point with all training points.

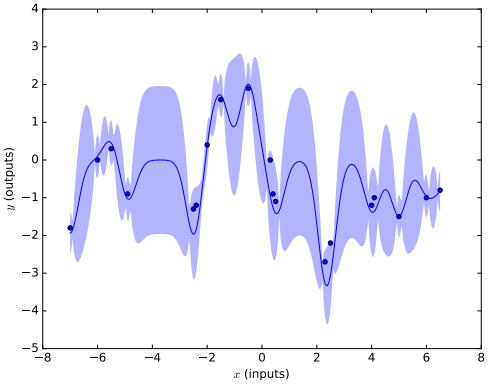
A few illustrative plots have been included below, with a Gaussian process performing inference on several data points but with different hyperparamters each time, to highlight the effect that they have on its predictive ability.



(1) $\sigma_f = 2.13, l = 1.27, \sigma_n = 0.17$



(2) $\sigma_f = 1, l = 1.7, \sigma_n = 0.1$



(3) $\sigma_f = 1, l = 0.3, \sigma_n = 0.001$

FIGURE 3.1. A set of synthetic plots were created to illustrate a Gaussian process's behaviour when different hyperparameters (as shown below each plot) are used to perform regression. The key things to observe in these 2D examples are that the noise and length scale govern the complexity of the model - both a low length length scale and error are required to create a Gaussian process that attempts to adhere to points as closely as possible, though this can result in 'overfitting' of the data, seems to be the case in plot 3.

3.2 Leave-One-Out Cross Validation

To determine the hyperparameters of the training data, cross-validation for model training is used, with the number of folds used, k, equal to the number of datapoints, only excluding one data point per round - hence the name. By optimising over the sum of cross-valiated log likelihoods, it is no longer strictly only assessing the log marginal likelihood, instead acting

as more of a pseudo-likelihood. Directly optimising over the marginal likelihood provides the probability of observed data *given model assumptions*, whereas the cross-validation approach provides the log predictive probability estimates independent of the fufilment of said model assumptions. (**?**) states in Chapter 4 that Gaussian cross validation methods are more robust to misspecification, including problems such as non-Gaussian errors. Cross-validation was chosen for this reason, as the intrinsic properties of the data were not studied in detail prior to performing experiments as a part of this study, and nor were biological experts consulted for the duration of the study to provide advice on the quality and soundness of predictive maps.

The log probability of the data ommitting training case i from the n input data points is:

$$\log p(y_i|X, \mathbf{y_{-i}}, \theta) = -\frac{1}{2}\log \sigma_i^2 - \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \frac{1}{2}\log 2\pi$$

for input data X, the $i-th$ response variable y, predictive mean μ, and predictive variance σ^2. The total log probability across all n subsets of the data of size $n-1$ is then:

$$L_{LOO}(X, y, \theta) = \sum_{i=1}^{n} \log p(y_i, X, \mathbf{y_{-i}}, \theta)$$

The LOO-CV predictive mean and variance can then be derived, and represented in terms of covariance matrix K, as calculated using Equation (3.2) and Equation (3.3).

$$\mu_i = y_i - \frac{[K^{-1}\mathbf{y}]_i}{[K^{-1}]_{ii}} \text{ and } \sigma_i^2 = \frac{1}{[K^{-1}]_{ii}}$$

To be able to optimise hyperparemeters over the feature dimensions more efficiently, the partial derivates over *each* is also needed:

$$\frac{\partial u_i}{\partial \theta_j} = \frac{[Z_j \alpha]}{[K^{-1}]_{ii}} - \frac{\alpha_i [Z_j K_{ii}^{-1}]_{ii}}{[K^{-1}]_{ii}^2}$$

$$\frac{\partial \sigma_i^2}{\partial \theta_j} = \frac{[Z_j K^{-1}]_{ii}}{[K^{-1}]_{ii}^2}$$

$$\text{where } \alpha = K^{-1}\mathbf{y} \text{ and } Z_j = K^{-1}\frac{\partial K}{\partial \theta_j}$$

With the means to perform Gaussian process regression by determining the hyperparameters of the kernel using leave-one-out cross-validation, the next step would be to apply this to Gaussian process classification.

3.3 Gaussian Process Classification

As benthic habitat mapping requires the prediction of discrete labels and not continuous values, Gaussian process regression is not directly applicable to the problem domain. Just as logistic regression can be formulated by post-processing of the results of a linear regressor, Gaussian process classification can be performed by using multiple Gaussian process regressors as its underlying model. The one-vs-all approach is taken in this case, requiring k separate Gaussian process regressors.

For example in Figure 3.2, when fitting a regressor for the green *class 1*, the labels at coordinates corresponding to label 1 are set to 1, and all other points to 0. For *blue* label 2, the labels of all the blue points are set to 1, and the rest 0, and so on so forth - this is applicable to any number of classes k, where the complexity increases linearly for each class that exists in the data.

When forming predictions, each of these separate Gaussian process regressors provide a different set of results as they consider separate one-vs-all cases as previously trained. For the k labels, and each regressor GP_k, where $i = 1, 2..., k$:

$$\bar{\mathbf{f}}_{*\mathbf{all}} = \begin{bmatrix} \bar{\mathbf{f}}_{*\mathbf{GP_1}} \\ \bar{\mathbf{f}}_{*\mathbf{GP_2}} \\ \dots \\ \bar{\mathbf{f}}_{*\mathbf{GP_{k-1}}} \\ \bar{\mathbf{f}}_{*\mathbf{GP_k}} \end{bmatrix}, \mathbb{V}[\mathbf{f}_*]_{\mathbf{all}} = \begin{bmatrix} \mathbb{V}[\mathbf{f}_*]_{\mathbf{GP_1}} \\ \mathbb{V}[\mathbf{f}_*]_{\mathbf{GP_2}} \\ \dots \\ \mathbb{V}[\mathbf{f}_*]_{\mathbf{GP_{k-1}}} \\ \mathbb{V}[\mathbf{f}_*]_{\mathbf{GP_k}} \end{bmatrix} \tag{3.6}$$

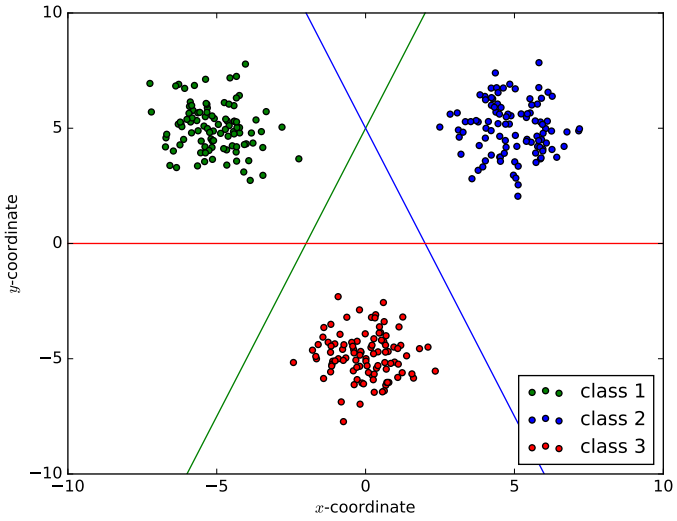


FIGURE 3.2. Simple example of data being split up to perform one-vs-all classification. The coloured lines corresponding to the label colours indicate the separation of clusters when performing each round of one-vs-all regression rounds. For class 3, everything below the red line is taken to be the 'positive label', and everything above, the 'negative label'. This is only illustrative, and the labels will just as easily be wrangled if they were interspersed instead of segregated like they are here.

As the original labels were changed to be constrained in the range $[0, 1]$, the same is done to the predictive means, by passing it through the logistic sigmoid function (**??**). The resulting matrix of predictive means and variances for any $k > 1$ provides a vector of k predictions for each input rather than the single label that classification would require. To simplify the probabilistic results per label at each point, the value with the highest probability (i.e. the argmax) would be taken, along with the matching variance containing the confidence intervals. As the $O(n^3)$ operations required for the matrix inversions for performing Gaussian process regression must be performed k times in total (once per label), the time required for anything more than several thousand points would make it impractical to use, requiring methods to sufficiently bring down the running time.

3.4 Gaussian Process Approximation

To be able to use Gaussian processes on larger data sets, approximations can be made to avoid paying the full cost of performing expensive operations, approximations can be made. The method of approximation that will be used in this study is ensemble methods. In the context of Gaussian processes, this involves breaking up the data into small chunks, where each 'expert' performs regression seperately, before taking the product of all the experts' results to give the approximation for the full dataset. The property that these ensemble methods have in common are that any given expert's decisions are weighted by their precision (inverse of variance - so a low variance would result in a high precision), such that experts that are more 'confident' in their results provide great input to the final weightings, whereas experts with low precision, or predictions that have a high variance, are unsure of their results, and hence provide comparably less input to the summarised predictions. The component that differs between the two approximations below (and these ensemble methods in general) is how the precision of each expert is used to weight their predictions.

3.4.0.0.1 Product of Gaussian Process Experts. The product of experts algorithm is a the simplest of Gaussian process ensemble methods, as the predictive variance of expert is directly used as-is to weight that particular expert's predictions, as shown in the equations below. The *product* of experts described above is proportional to a single Gaussian process with the following product of expert (PoE) predictive mean and variance:

$$\mu_*^{poe} = (\sigma_*^{poe})^2 \sum_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{3.7}$$

$$(\sigma_*^{poe})^{-2} = \sum_k \sigma_k^{-2}(\mathbf{x}_*) \tag{3.8}$$

where $\sigma_k^{-2}(\mathbf{x}_*)$ are the predictive Gaussian precisions (inverse of Gaussian variances), and $\mu_k(\mathbf{x}_*)$ are the predictive Gaussian means. Equation (3.8) describes the precision of the product of experts model in its entirety, by summing up the Gaussian precisions of all of the experts. For each point, this means that its PoE variance will be low if every expert calculates a low predictive variance, whereas if a sufficient number of them end up predicting a high variance

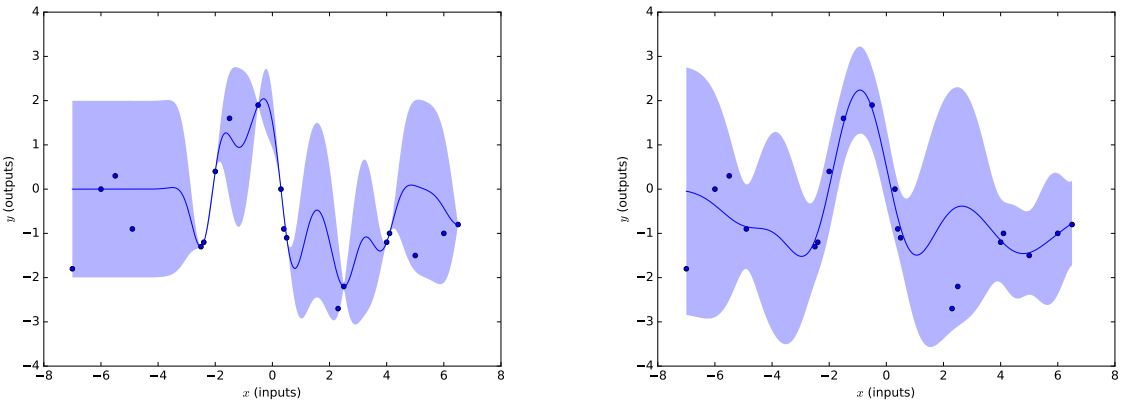
for a given point, the overall PoE variance for that point will likely be high. This high variance is then applied to the PoE predictive means at every point, where the the product of each *individual* expert's Gaussian precisions and Gaussian means are taken, and summed with all the other experts' at the same point, giving a higher precision ($\sigma_k^{-2}(\mathbf{x}_*)$) a higher weight.

3.4.0.0.2 Generalised Product of Gaussian Process Experts.

$$\mu_*^{gpoe} = (\sigma_*^{gpoe})^2 \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{3.9}$$

$$(\sigma_*^{gpoe})^{-2} = \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*) \tag{3.10}$$

The only difference in generalised product of Gaussian experts (GPoE) is the use of β, where its purpose is to allow the explicit weighting of experts based on the circumstances. The value of each β_k is flexible, but as scaling Gaussian processes to large datasets isn't the primary focus of this study, we simply set each β_k to $\frac{1}{M}$, where M is the number of experts, as suggested in (Deisenroth, 2015) to be able to maintain reasonable margins of error. To show how they perform, below is a performance comparison between these two ensemble methods on a simple dataset:



Product of Experts Generalised Product of Experts

FIGURE 3.3. Examples of Gaussian process ensemble methods. As a result of the β in the GPoE scaling the variances, they no longer cancel each other out when summed together, explaining the comparatively larger variance in the GPoE compared to the PoE, even when predictions are very close to the correct values. While these conservative (wide) variances can be a negative aspect as mentioned in Deisenroth (2015), it may also prove to be beneficial, as the experiments later show.

3.5 Summary

In this section, we explored the probabilistic capabilities of Gaussian processes regression, and how this translates into Gaussian process classification. As these presented very restrictive limits on the size of data that can be worked with, methods to estimate them by breaking down the algorithm into emabarrasingly parallelisable chunks were explored, along with how they compare in terms of performance. It would amiss at this point not to address the existence of multi-output Gaussian processes that can work with correlated data (as the data in benthic habitat mapping would be - the habitats that co-eixst do not do so in isolation and no relation to each other), as explored in works such as (cite a few works). However, the multi-output that they deal with is over arbitrary ranges. Because in this particular domain, normalised predictions per point must sum to 1 (or the original total label count at a point, in the case of training data), multi-output Gaussian processes do not enforce this constraint, and as such, their use was not explored in this study. Without a way to correctly model the multi-output data using Gaussian processes, other methods that are able to need to be explored.

CHAPTER 4

Multi-output Habitat Mapping

On top of being able to produce probabilistic outputs, it would be of further advantage if predictions could be performed on multi-output data as well, to fully utilise the fact that many areas of the benthos will contain more than one label at any given time, where the simplifcation of these multi-labels to a single one thus far causes a considerable loss of information from the original data before the model fitting even begins. This is the motivation for us to explore Dirichlet multinomial distributions that have the ability to perform predictions over category counts, a perfect fit for the original data collected in this study.

Dirichlet multinomial regression, as the name suggestions, combines dirichlet and multinomial distributions to achieve the combined model. In particular, we are interested in modeling a distribution over category counts, as there exists relationship in our data such that every bathmetry point corresponds to a certain count of each possible label in the relevant area of benthos. To appreciate the Dirichlet Multinomial distribution as a whole, we briefly and describe the multinomial and Dirichlet distributions and the relationship, before delving into the equations needed to perform Dirichlet mulitnomial regression.

4.1 Multinomial Distribution

Multinomial distributions are the generalisation of binomial distributions that describe the probability of, for example, getting a heads or tails when flipping a two-sided coin. Generalising beyond 2 categories, the multinomial does the same, but for any integer n number of categories, like throwing an n-sided weighted die (weighted for the problem at hand). The relevance to the multi-label habitat data is then quite apparent, given that each data point needs to be mapped

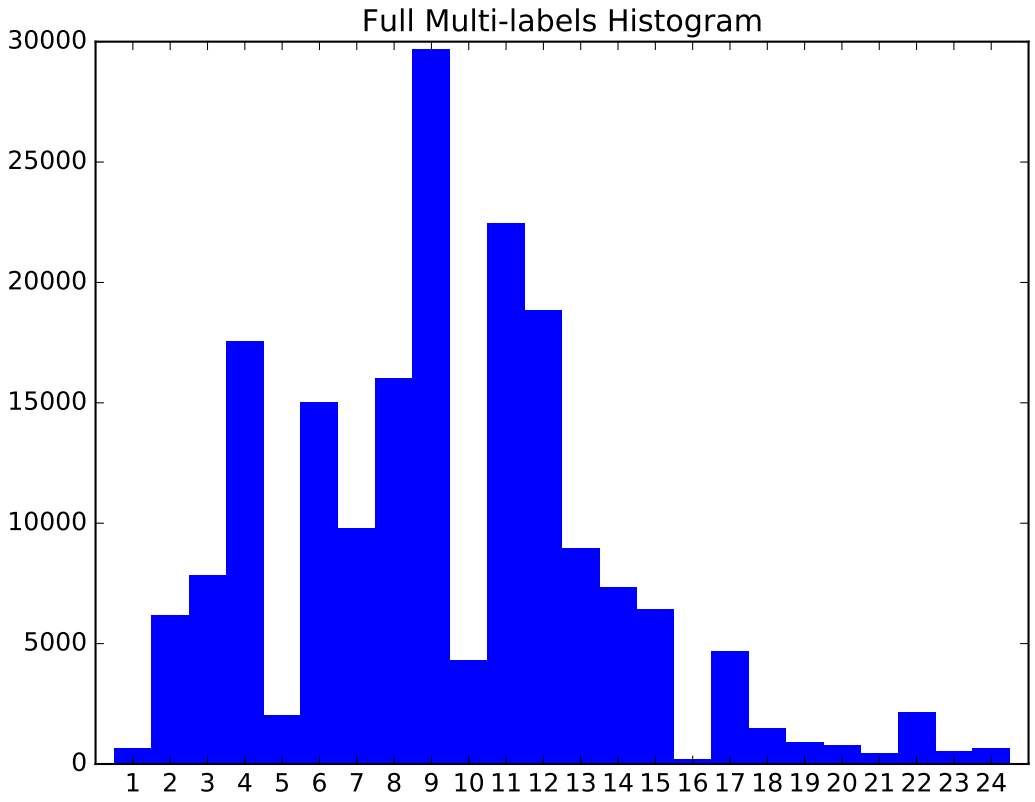


FIGURE 4.1. Histogram of label counts for every multi-label data point

to any number of K categories and sum to some number of samples stated. Draws from a multinomial distribution can be represented using a histogram such as that in Figure 4.1. However, this particular histogram encapsulates the label distributions for the entirety of Scott Reef instead of what we actually want - a single coordinate.

4.2 Dirichlet Distribution

To be able to perform predictions on individual points, it would seem that draws from a multinomial distribution need to be doable for a specific coordinate alone - but there does not exist the information to do so using the multinomial distribution alone, as a single vector of counts/-draws as present in the multi-label data would be vastly insufficient. This is where the Dirichlet distribution comes in, placing a prior over the possible multinomial distributions. Parameters

of the multinomial distribution can then be drawn from a Dirichlet given α, the concentration parameters that govern the distribution of multinomial coefficients that can be drawn. The actual representation of a Dirichlet are K-dimensional vectors where K is the number of possible categories where $\sum \mathbf{x} = 1$ for any vector $\mathbf{x}$, representing the distribution of K labels that form the exhaustive set of possibilities, and hence sums to 1. When working in 3 dimensions, this can be represented visually on a triangle anchored at the unit coordinates along each of the x, y, z axis (Figure 4.2).

Given that the aim is to draw parameters for multinomial distributions from our Dirichlet, the following model is considered, where vectors $\mathbf{p}$ are the multinomial coefficients:

$$\mathbf{p}|\alpha = (p_1, \ldots, p_k) \sim \mathrm{Dir}(K, \alpha) \tag{4.1}$$

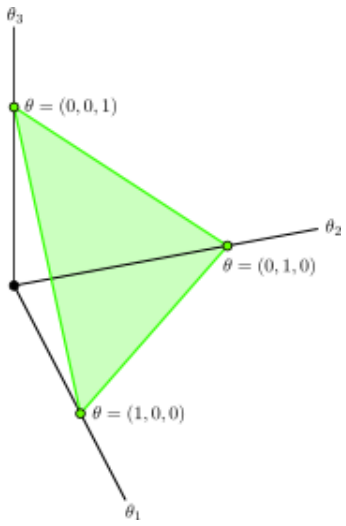
The last important aspect of the Dirichlet distribution that needs to be looked at before moving onto the next section where the Dirichlet multinomial regression explained more formally, is calculating its entropy. Given the benefits of probabilistic output described in Chapter 3, it would only make sense for a Dirichlet distribution to be able to provide simiilar information for vectors containing distributions over categories. This is done via its entropy:

$$\log \mathrm{B}(\alpha) - (k - \alpha_0)\psi(\alpha_0) - \sum_{j=1}^{k}(\alpha_j - 1)\psi(\alpha_j) \tag{4.2}$$

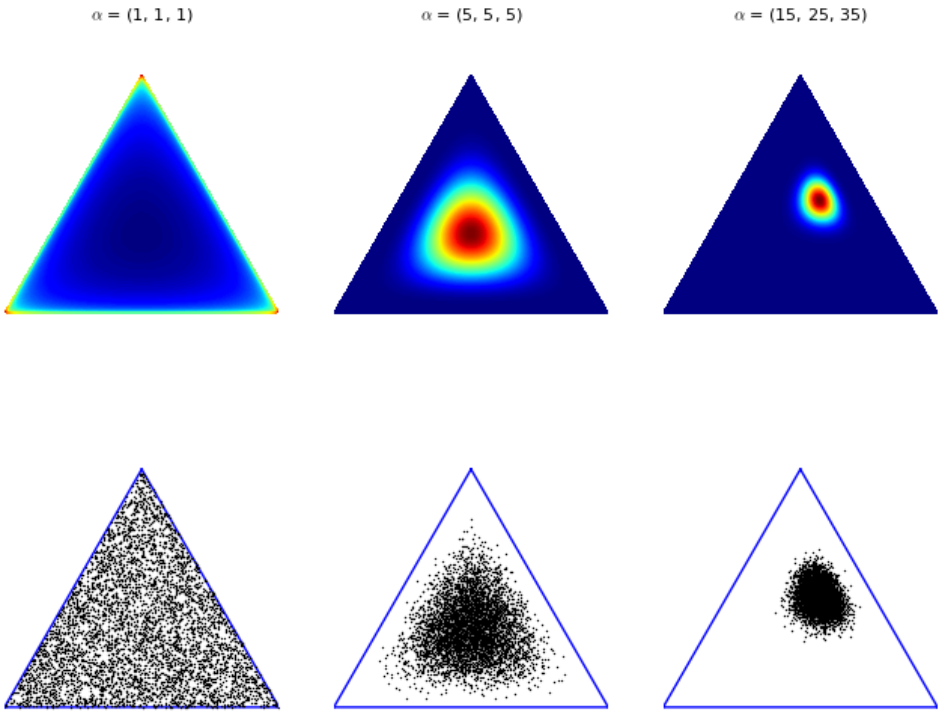
where

$$\mathrm{B}(\alpha)\frac{\Pi_{i=1}^{k}\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k}\alpha_i)}, \text{ and } \alpha_0 = \sum_{i=1}^{k}\alpha_i \tag{4.3}$$

This would produce an entropy value for every vector of normalised category counts produced by the Dirichlet multinomial that can then be displayed using a heatmap over the same coordinates with query data for Scott Reef, providing a visual representation of entropy to allow easy identification of low entropy areas (higher confidence in predictions), compared to high entropy areas (higher variance in predictions and hence lower confidence).



Simplex plane in 3-dimensional space



These three simplexes show the distributions of the multinomial coefficient draws for different values of α, with the top showing the density of points using a heatmap, and the bottom plotting those points directly. When $\alpha = 1$, the Dirichlet is equivalent to the uniform distribution. As α increases, the density of points towards a particular direction increasesas can be seen especially when comparing the second and third simplexes.

FIGURE 4.2

4.3 Dirichlet Multinomial Regression

Equipped with some basic knowledge about the Dirichlet and multinomial distributions, the steps involved in Dirichlet multinomial regression can be explained in context. The previous two sections looked at the two distributions in isolation - the aim now is to combine them in the Dirichlet multinomial, then optimise the parameters to be able to predict normalised category counts given training data, as previously described. The following formula is the joint prior distribution:

$$\mathrm{DirMult}(C|\alpha) = \frac{M!}{\Pi_k C_k!} \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\sum_k c_k + \alpha_k)} \Pi_{k=1}^{k} \frac{\Gamma(C_k + \alpha_k)}{\Gamma(\alpha_k)} \tag{4.4}$$

where C are the category counts, α are the Dirichlet parameters as described in Section 4.2, and $M = \sum_k c_k$

The crucial part that has not yet been explained is how α is obtained/tuned to fit any training data available. In this study, the activation function chosen to calculate the α values was the softmax, also referred to as the 'normalised exponential', in that it has better numerical stability than the exponential function (plot example if time allows). The softmax generalises the logistic function to K-dimensional data like that being used in this study:

$$\alpha_k = \mathrm{softmax}\, \mathbf{X}^T \mathbf{w_k} \tag{4.5}$$

where

$$\mathrm{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \text{ for } i = 1, \ldots, K \tag{4.6}$$

The weights w here are in fact a matrix of weights with dimensions $(K \times D)$, where K is the number of possible labels across the dataset, and D is the dimensionality of the dataset. Muliplying the dirichlet multinomial prior by the Gaussian likelihood over weights then then gives the posterior that needs to be optimised to obtain the weights required to predict the normalised label counts at any given point:

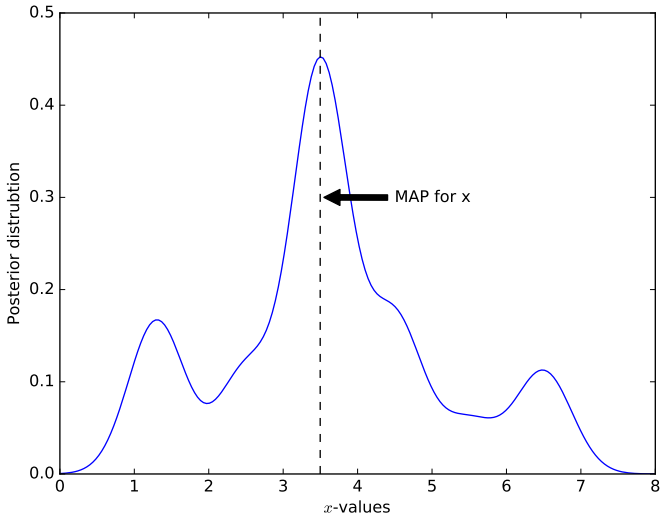


FIGURE 4.3. Simple example of the MAP value for the posterior distribution over some variable x

$$\Pi_{n=1}^{N} \text{DirMult}(c_n|\alpha(x_n)) \times \Pi_{n=1}^{k} \sim \mathcal{N}(w_k|0, \phi I) \tag{4.7}$$

where ϕ is a regulariser that governs how much the weights w are allowed to vary during the fitting process. To be able to optimise the weights, maximum a posteriori (MAP) estimation is used, which takes the set of parameters at the most likely point over the distribution. For example, in the 2-dimensional example in Figure 4.3, the dotted line represents the value of x that maximises the posterior distribution $P_{X|Y}(x|y)$.

To perform MAP on the posterior distribution P, we take the log of P, as this scales the data in a way that allows optimisation algorithms working in multiple dimensions to determine how to search the space more efficiently:

$$\begin{aligned}\log(P) = \sum_{n=1}^{N}[\log(M_k) - \sum_{k}\log(c_k!) + \log\Gamma(\sum_{k}\alpha_k(x_n)) - \log\Gamma(\sum_{k}c_{nk} + \alpha_k(x_n))] \\ + \sum_{n=1}^{N}\sum_{k=1}^{K}[\log\Gamma(c_k + \alpha(x)) - \log\Gamma(\alpha_k(x_n))] \\ + \sum_{k=1}^{K}[-\frac{\phi}{2}\log(2\pi\phi) - \frac{1}{2}w_k^T\phi\mathbb{I}w_k]\end{aligned} \tag{4.8}$$

To optimise this equation, the partial derivative of the above over the weights w are considered:

$$\partial \frac{\log p(c,x)}{\partial w_k} = \sum_{n=1}^{N} x_n \alpha_k(x_n)[\psi(\sum_l \alpha_l(x_n)) - \psi(\sum_k c_{nk} + \alpha_k(x_n))] + \sum_{n=1}^{N} x_n \alpha_k(x_n)[\psi(c_{nk} + \alpha_k(x_n)) - \psi(\alpha_k(x_n))] - \frac{1}{\phi} w_k \quad (4.9)$$

Given the log posterior and its partial derivatives over every weight, the MAP can then be calculated using parameter optimisation algorithms available in machine learning libraries (introduce this briefly the first time it comes up, in the GP section, then just to refer to that here.).

4.3.1 Using Markov Chain Monte Carlo to Sample Weights

Given the distribution of possible values of the weights $\mathbf{W}$ described, taking only the MAP for these values would be discarding other information in the distribution that also describes the data. While the MAP allows a single set of weights to be determined that allows performance to be assessed in addition to plotting maps to visualise the distribution of each habitat class in separate heatmaps, the 'correctness' of the MAP estimation can further be assessed by generating maps created from weights from throughout the posterior distribution as they are drawn, then observing how consistent certain areas of the map remain when doing so.

To draw weights from the posterior distribution, Markov Chain Monte Carlo (MCMC) can be used. MCMC involves 'random walks' through multiple dimensional space when using the Metropolis Hastings algorithm, where every step in n dimensional space has a chance to be rejected or accepted. To account for the randomness in these walks (that provide *chains* of parameters that the posterior is concerned with), Gelman-Rubin's r-hat statistic can be used to measure convergence when multiple MCMC instances are run for the same posterior. For multi-dimensional data, we will simply calculate the r-hat value for each dimension (each $w_{ij} in \mathbf{W}$ for $i = 1, \ldots, K, j = 1, \ldots, D$), and take their average.

(may need to explain MCMC and rhat in at least a little more detail here. at least include equation for r-hat statistic used)

4.4 Illustrative Example

The differences between a Gaussian Process that provides the probability distribution of possible labels compared to the Dirichlet Multinomial Regressor that provides the distribution of actual labels at a point, are highlighted in the illustrative example below. Three clusters A, B, C, over two labels were generated, such that B, C preferred labels $2, 1$, respectively, while cluster A contained an even mix of both - this can be observed by matching the colours of clusters to the adjacent colour bars.

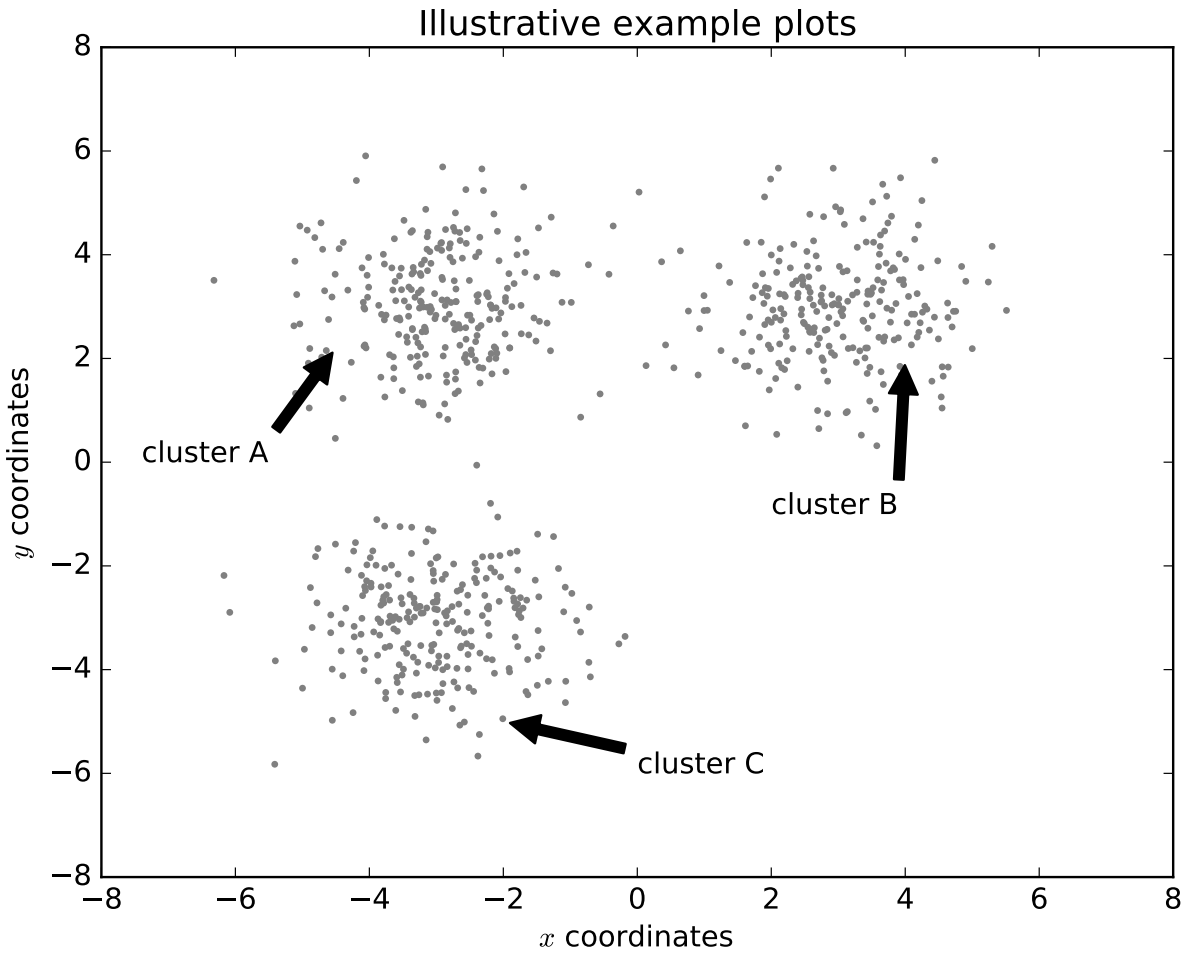
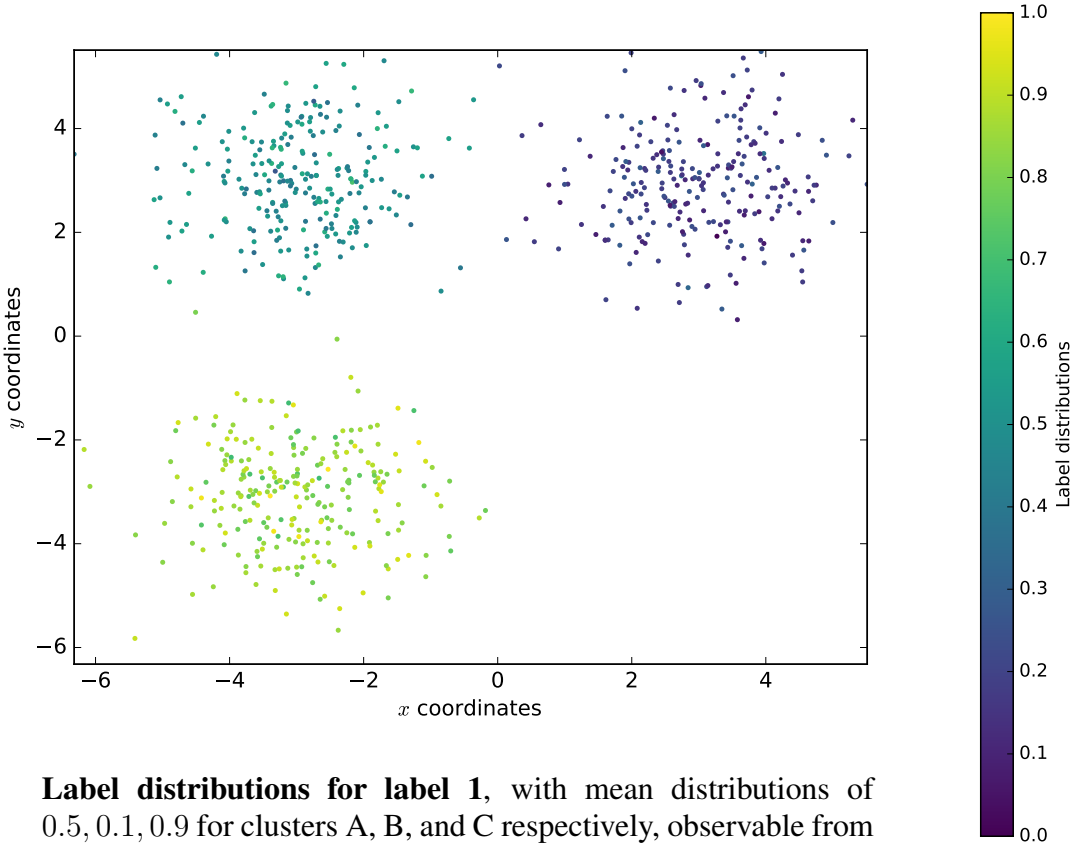
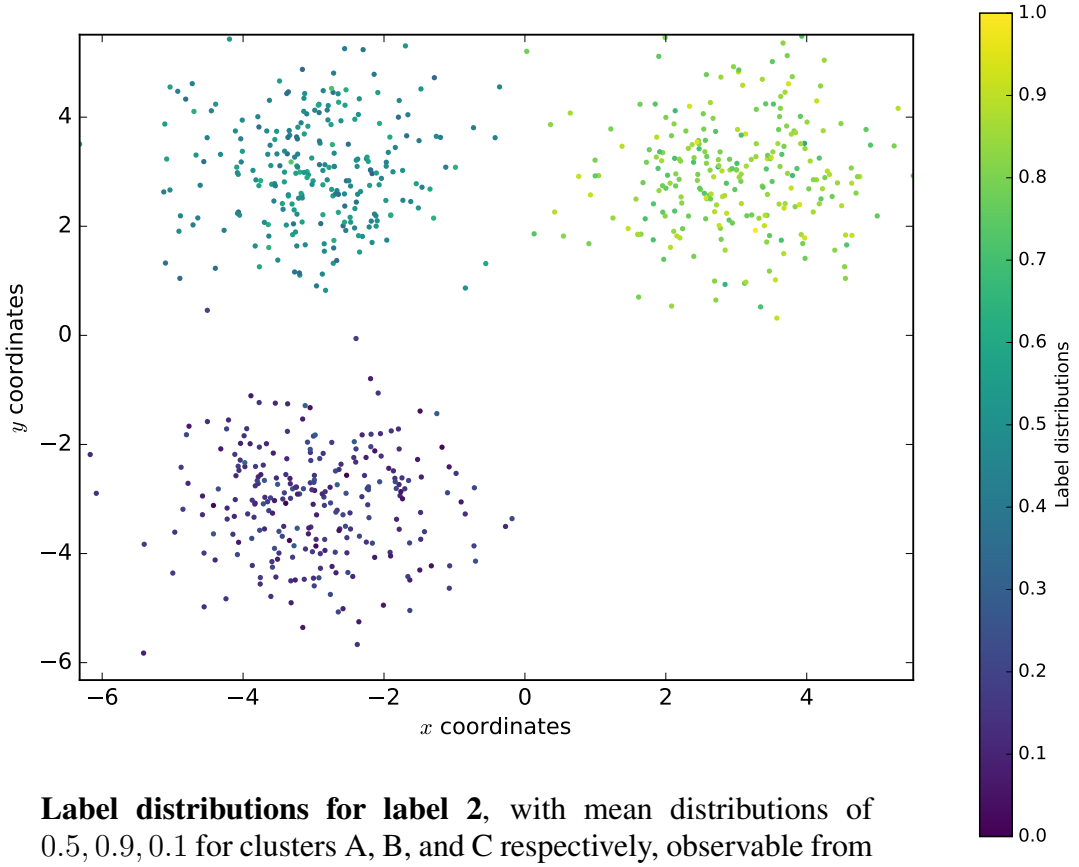


FIGURE 4.4. Plot of the synthesised data used. Only the (x,y) coordinates of the points are shown here, with actual label distributions shown below.



Label distributions for label 1, with mean distributions of $0.5, 0.1, 0.9$ for clusters A, B, and C respectively, observable from the colour bar.



Label distributions for label 2, with mean distributions of $0.5, 0.9, 0.1$ for clusters A, B, and C respectively, observable from the colour bar.

In this example, the GP and DM models were each trained on half of each cluster, and made to predict the other half. However, as a standard GPC can only have single label inputs and outputs, a approximation/simplification was made for the purpose of calculating average error, whereby the label was simply taken to be the most frequently occurring label at any given point. While this is a reasonable simplification for clusters A, B as the dominant label has majority share, this is not the case for C, as the split between the two labels per point in the cluster is exactly even.

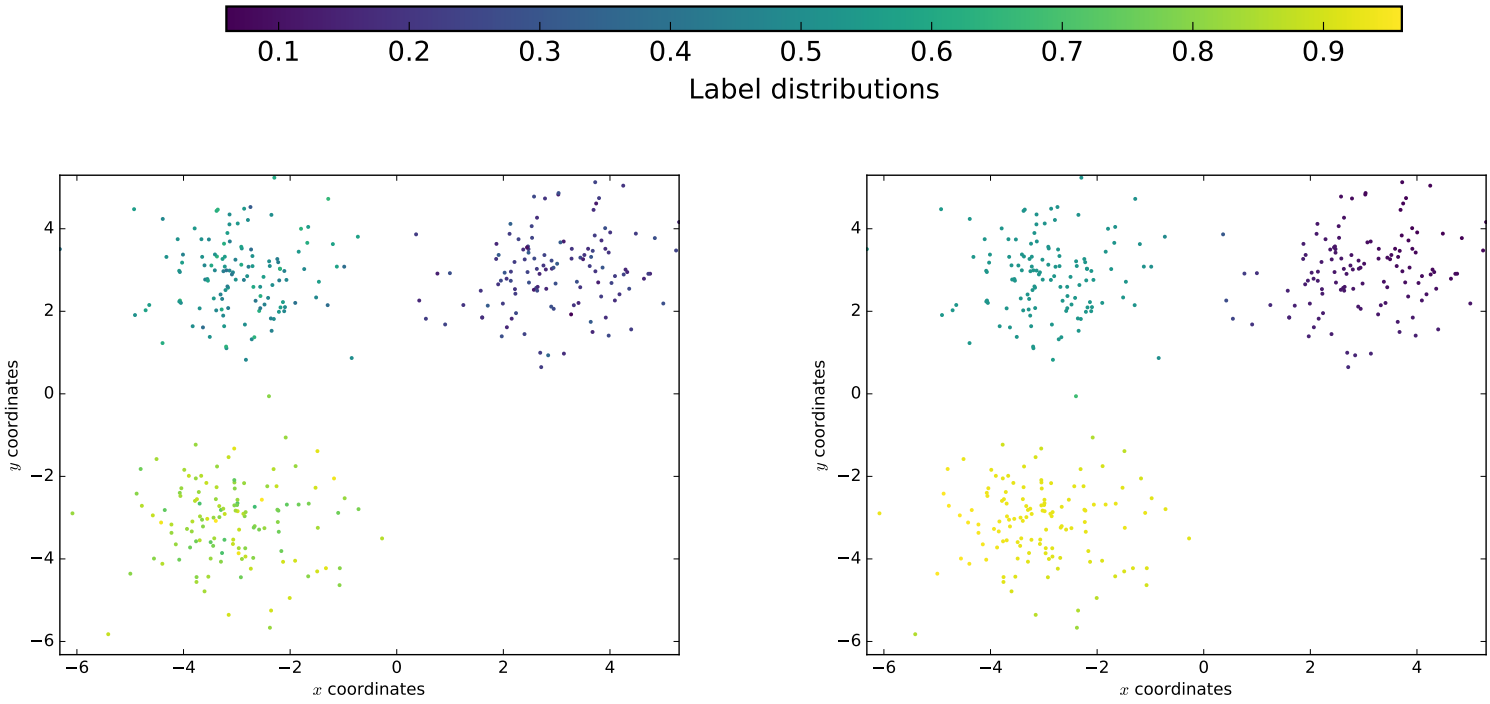
4.4.1 Results

The results in Section 4.4.1 were taken from an average of 20 runs. They show that the DM performed best when projecting the data to quadratic space, while the GPC did best on the original data as-is. This was taken into account for the plots below for the DM and GP respectively, which used an instance of the more favourably performing processed data. The DM plots below show the actual distributions for each label, allowing comparison with the true values, but the GP has taken the argmax of every label instead, as it cannot represent multi-label data.

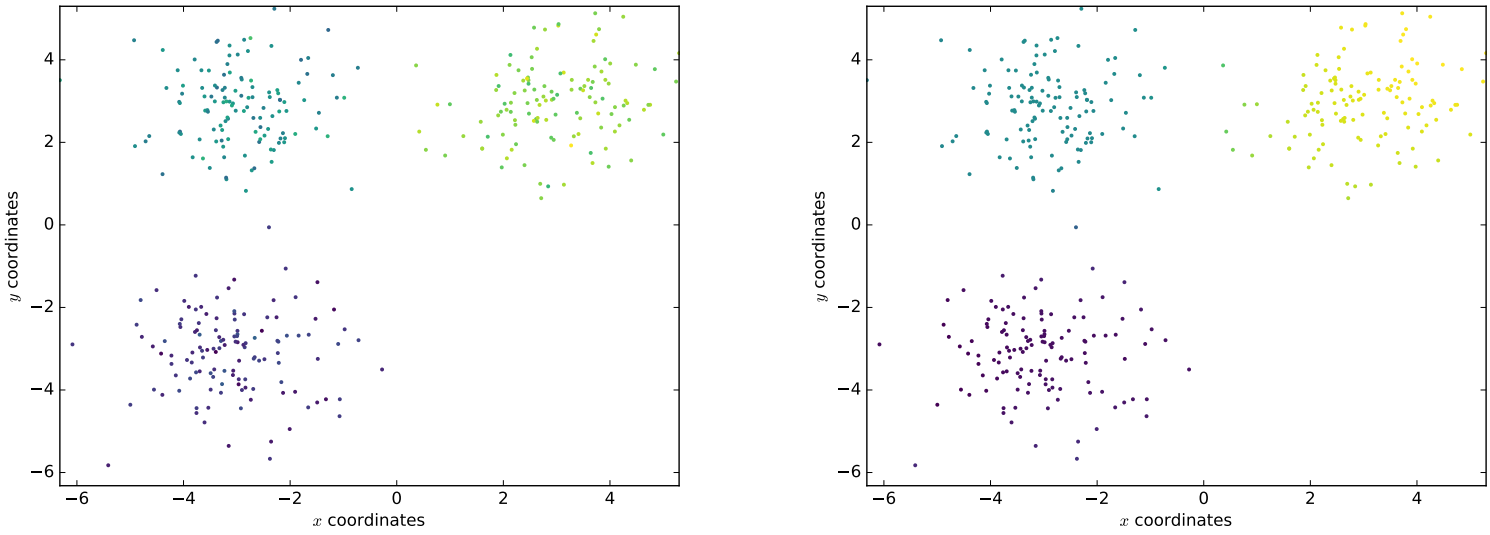
	Dirichlet Multinomial Regression RMSE*	Gaussian Process Classifier (argmax) RMSE
Original data	0.0702	0.2683
Quadratic-space projection	0.0656	0.4343
Cubic-space projection	0.2902	0.4373

RMSE - root mean squared error

Figure 4.5 shows that the DM was able to predict values close to the distributions in the original full clusters, with each the predictions in each clusters A, B with errors of 0.045, while C increases slightly to 0.052. The Gaussian process classifier was less consistent, however. Its accuracy in predicting the correct (argmax) label across all three clusters was 0.77, where cluster A achieve an accuracy of 0.99, cluster B 0.92, but cluster C sharply dropped off to 0.416 - given that this illustrative example only contains 2 classes, this corresponds to a predictive ability worse than random guess.

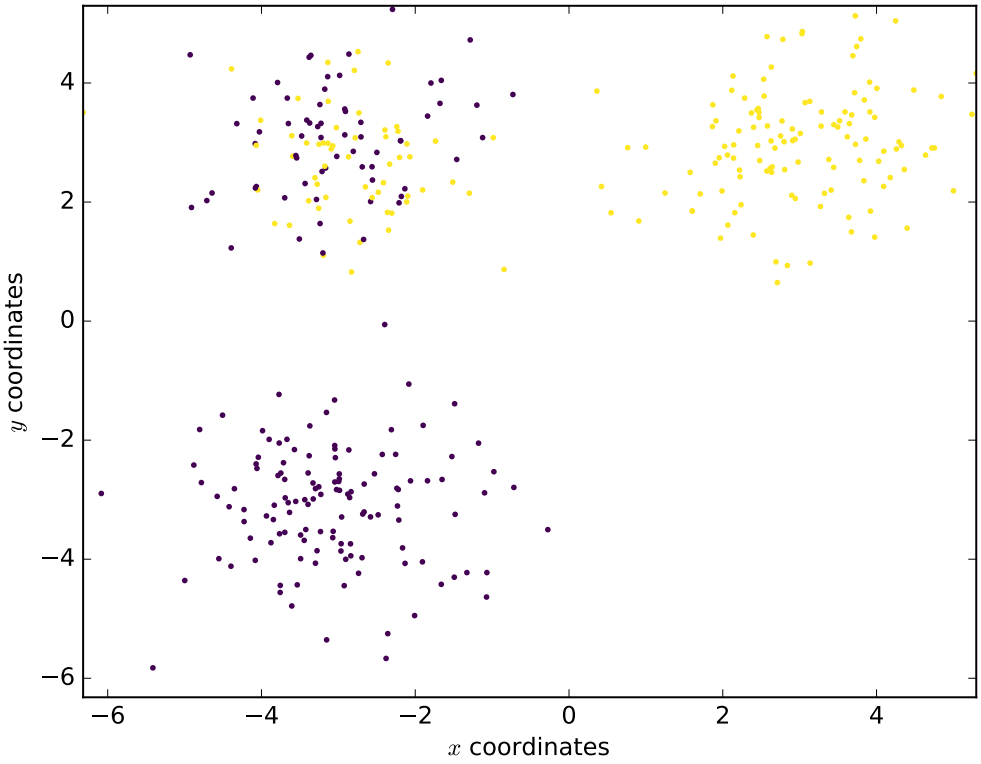


Label 1 - expected (left) and predicted (right) label distributions.

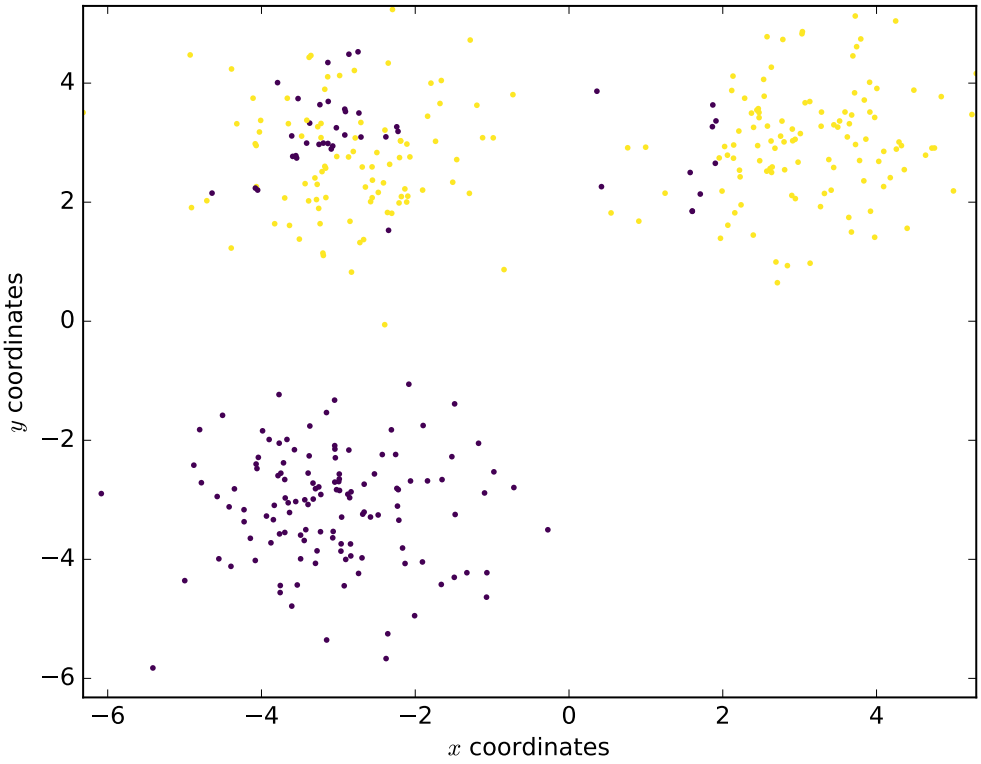


Label 2 expected (left) and predicted (right) label distributions

FIGURE 4.5. Coloured scatter maps of Dirichlet multinomial predictions. Where the yellow/green regeions appear to point to a discrepancy in predictions, it would be useful to refer to the colour bar to observe that yellow (≈ 0.9)) is near to green, the lighter shades in particular ($0.75 \sim 0.85$)



Expected labels from taking the most probable label per point



Gaussian process predictions after taking the most probable label oer data point.

FIGURE 4.6. Gaussian process expected and predicted scatter maps

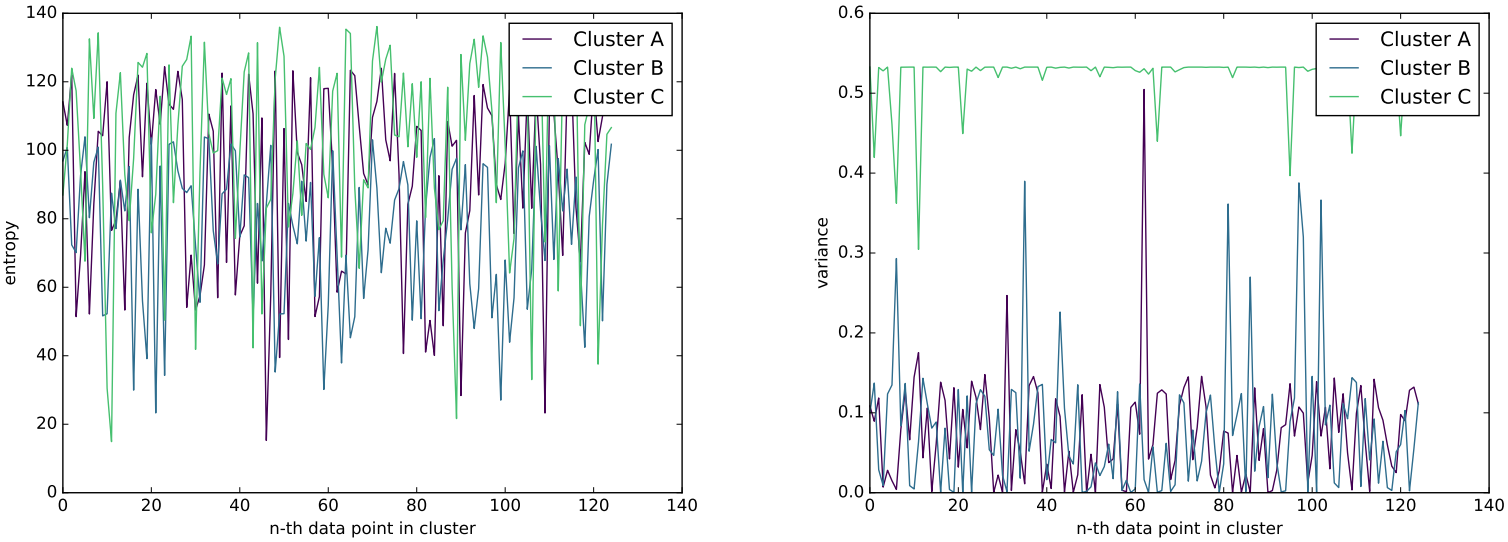


FIGURE 4.7. On the **left** are the plots of the **Dirichlet multinomial entropies** in clusters A, B, and C as indicated by the legend, and the same thing on the **right** for the **Gaussian process variances** (for simplicity, the the average of the two variances over each of the classes was taken). It is immediately noticable that while clusters A, B had low variance most of the time, cluster C's variance constantly remained unreasonably high. In comparison, the trend of the entropies in cluster C is higer than in A and B, but only by a small amount, and not remotely to the extent of the Gaussian proceses' variances.

The Gaussian process' variance in the third cluster reinforces this fact - whereas the average variances in clusters A, B were only 0.076 and 0.077 respectively, it was 0.522 in cluster C. On the other hand, the DM's confidence of a mixture of labels remains relatively stable (with a slight increase in entropy, however, with the average entropies in each cluster being $92, 87,$ and 101 respectively.

From this basic example, it is apparent that in the area where there is an even mix of labels A, B, the Gaussian Process' predictions are both noisy and very uncertain about their predictions, where human intervention would be required to observe the fact that it is in fact a consistent mix of both. In contrast, the dirichlet multinomial regressor is able to remain relatively confident of the fact that an area does in fact have a mix of labels.

CHAPTER 5

Experiments and Results

To show that using Dirichlet Multinomial Regression provides richer and more valuable information than single-output or deterministic methods alone, we ran experiments on the data obtained from the ACFR's Sirius AUV and Schmidt's Falkor. In this chapter, we first assess the performance and usefulness of information of single output labels, to first highlight the need for models that can effectively perform multi-output predictions.

As seen in the previous chapter, a key benefit to applying the Dirichlet Multionomial distribution to the data is that it able to naturally perform multi-output predictions on label distributions that correctly sum back to 1. To illustrate this point, SVMs, Linear Regression, K-Nearest Neighbour, Random Forest, and Gaussian Process Regression were all coerced to perform multiple predictions across each label's normalised distribution values, where the results were compared with those of a Dirichlet Multinomial Regressor. However, an important point to keep in mind is that these models do not maintain the constraint of predictions per point summing back to 1, but they have been included in the experiments to illustrate if they can still provide reasonable results despite being an inherently 'incorrect' model, with the advantage of having implementations more readily available in open source libraries, and being more exhaustively studied in literature in general.

We also explored how to use the data to extract information about biodiversity and the corresponding confidence, indicated by the predictive variance in the case of Gaussian Processes and Dirichlet Multinomials. Moreover, to contrast the Dirichlet Multinomial's ability to naturally provide information about co-existing habitats with certainty, we compared the regions in which the Dirichlet Multinomial was certain with those in Gaussian Process predictions,

looking at both the overlapping areas and the corresponding level of variance observed in both models.

5.1 Training Data

To perform our experiments, bathymetry data and images of Scott Reef Central were used (Figure 5.1). The bathymetry data was collected using Eric Schmidt's Falkor a ship dedicated to marine research, with the depth for a large portion of the reef collected (imshow of bathy depth). Over 700GB of 'truthing' image data was collected by The University of Sydny's Australian Centre for Field Robotic's Sirius autonomous underwater vehicle (AUV). The training set provided already had labels assigned, which was a result of previous efforts using Variational Dirichlet Processes that performed the unsupervised clustering (Steinberg et al., 2011).

On close inspection, the UTM[1] coordinates in the training set do not correspond to the original data available from (for Field Robotics , ACFR) - this was because the exact point of retrieval for the bathymetry and image were not exact matches. To account for this, labels corresponding to bathymetry points were in fact taken from the closest images, rather than exact longitude/latitude or UTM matches, although the UTM coordinates in the training data itself remains as the original. Due to data wrangling previously done by Asher Bender, features of rugosity (roughness) and slope are also available, each measured at resolutions of 2m, 4m, 8m, and 16m.

[1]UTM is short for Universal Transverse Mercator, a coordinate system that splits the Earth up into a grid-like structure, where a location is given by the grid key, then an (x, y) coordinate in metres that defined a position from the 'origin' of the given grid.

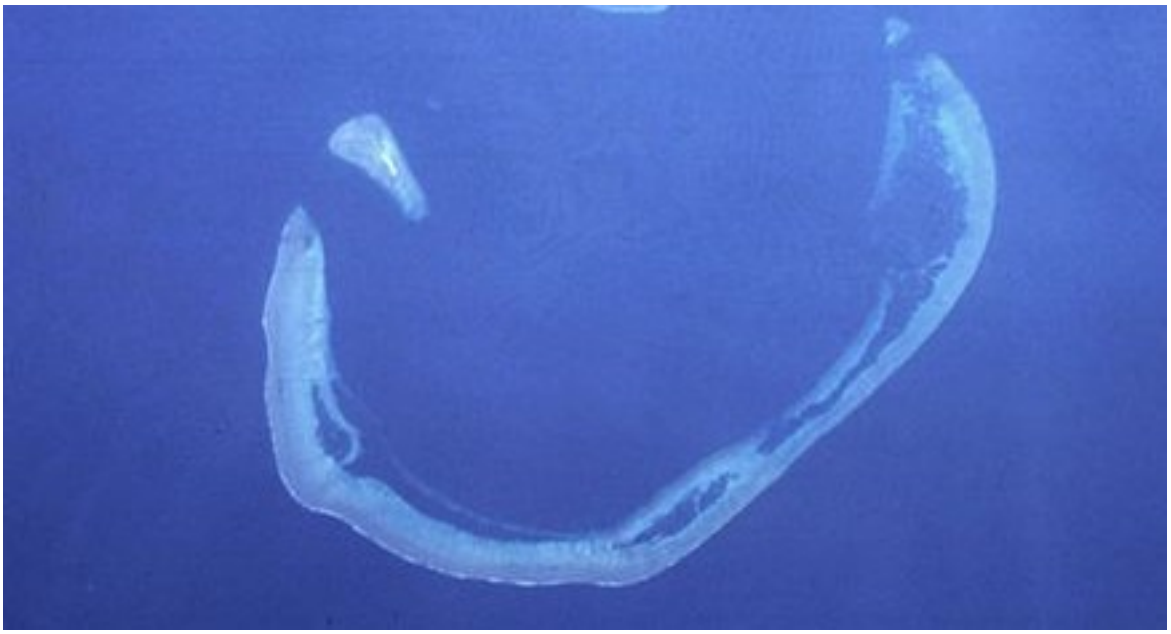


FIGURE 5.1. Aerial shot of Scott Reef from (National Aeronautics and Space Administration(NASA), 1996)

5.2 Data Preprocessing

5.2.1 Downsampling the Data

As the purpose of using Dirichlet Multinomial Regression was to be able to model the distribution of habitat label occurrences over an area, we downsampled the combined 2011, 2015 dataset which was at a significantly higher resolution than the 2009 dataset(cite comparison with paper using low resolution 100m data in contrast to 20 here as 'low', can't find atm). To do this, a 'grid' of squares was mapped onto the original space, and points in each grid grouped together into a single point. For the multi-output labels, all label counts in each lower-resolution grid were summed, whereas for the single label case, the label with the highest count in the downsampled data was taken to be the 'correct' label at that location.

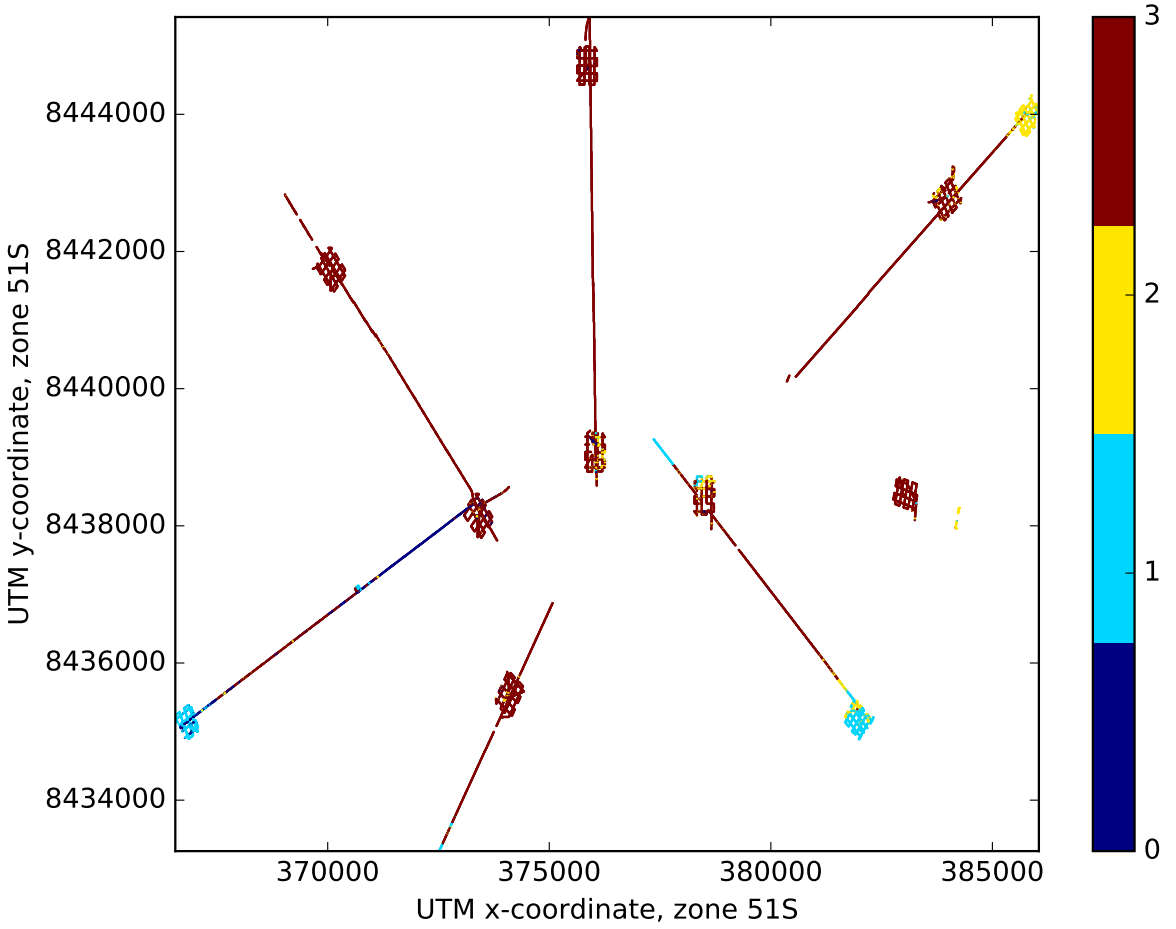


FIGURE 5.2. Original training data points (16502)

5.2.2 Simplifying labels

Another preprocessing step that was performed was the aggregation of habitat labels. The original training data contained 24 separate labels (Figure 5.3), determined through an automated clustering procedure using Dirichlet Processes on the raw images collected. Because of the uneven distribution of these labels (Figure 5.7 and Figure 5.8), with the occurrence of some too insignificant for any machine learning algorithms to pick up, they were simplified in collaboration with ecological experts, who manually identified which of the 24 labels were in fact of the same class - for example, 5 separate classes of coral may have been indistinguishable

to the average person, and were hence grouped into a single label. This allowed the near-non-occurring labels to be grouped together with more commonly occurring ones, whilst also allowing a different level of granularity in training models/forming predictions that could be used if only an approximation equivalent to observable human differences of an area's benthic map were required. This brought the number of labels in the simplified case from 24 down to 4 (Figure 5.4).

simplified	original
0 - Coral	1, 2, 18, 20, 21, 23, 24
1 - Rhodlidths	3, 5, 10, 16, 17, 19, 22
2 - Sand	13, 14, 15
3 - Halameda	4, 6, 7, 8, 9, 11, 12

TABLE 5.1. Full-simplified label mappings label mappings - sand, coral, patchy coral, (?) halameda, rhodliths

Note that from this point onwards, we will be working with the reduced feature set, in line with the aim of the paper to show the advantages of dirichlet multinomial regression when studies (environmental or otherwise) are limited to lower resolution data where strictly assigning only a single label to the features at a given data point is not representative of the otherwise rich information available. This restriction is a realistic one, because to be able to monitor large portions of the ocean for conservational and management reasons amongst others, data needs to be collected economically en-masse - and this means not collecting very high resolution data that would attract large costs at scale.

5.2.3 Preprocessing and Feature Projection

To maximise performance of the algorithms used across the experiments, a number of preprocessing steps were taken to improve the predictions made. The features in the data were first scaled, where each feature was centred to the mean with unit variance), then normalised over each future such that they had unit length. To allow the algorithms tested to learn the data and its complexities better, projecting the data into higher dimensional space was required. Full quadratic projections ($x_0, x_1, x_2 \Rightarrow x_0^2, x_1^2, x_2, 2x_0x_1, 2x_0x_2, 2x_1x_2$) and squared terms with a 1 bias terms ($x_0, x_1, x_2 \Rightarrow x_0 + x_1, x_2, x_0^2, x_1^2, x_2^2, 1$) were both tested. The latter was chosen, as

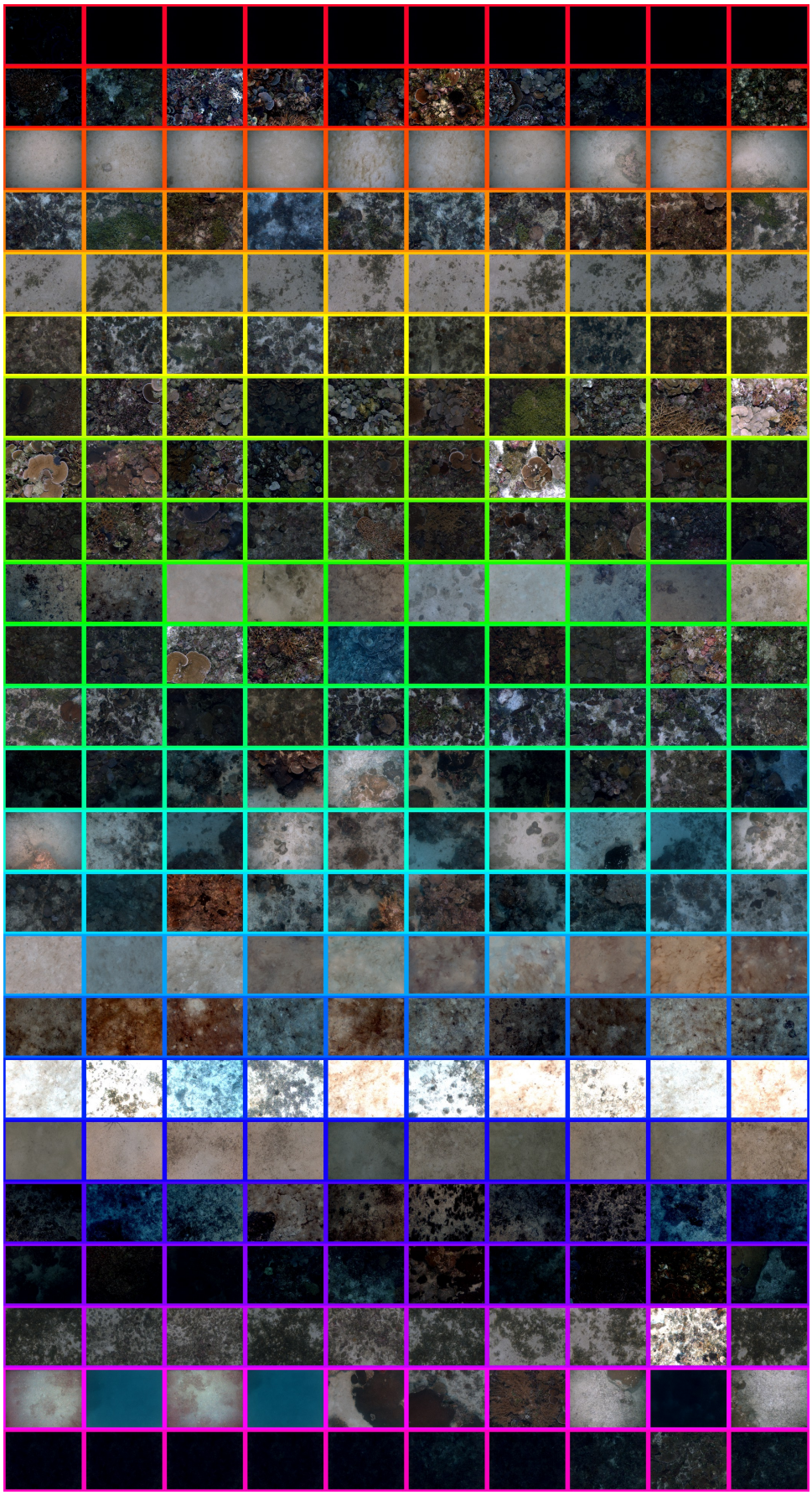


FIGURE 5.3. Samples of images from each of the full 24 classes mark with the simplified labels adjacent to it

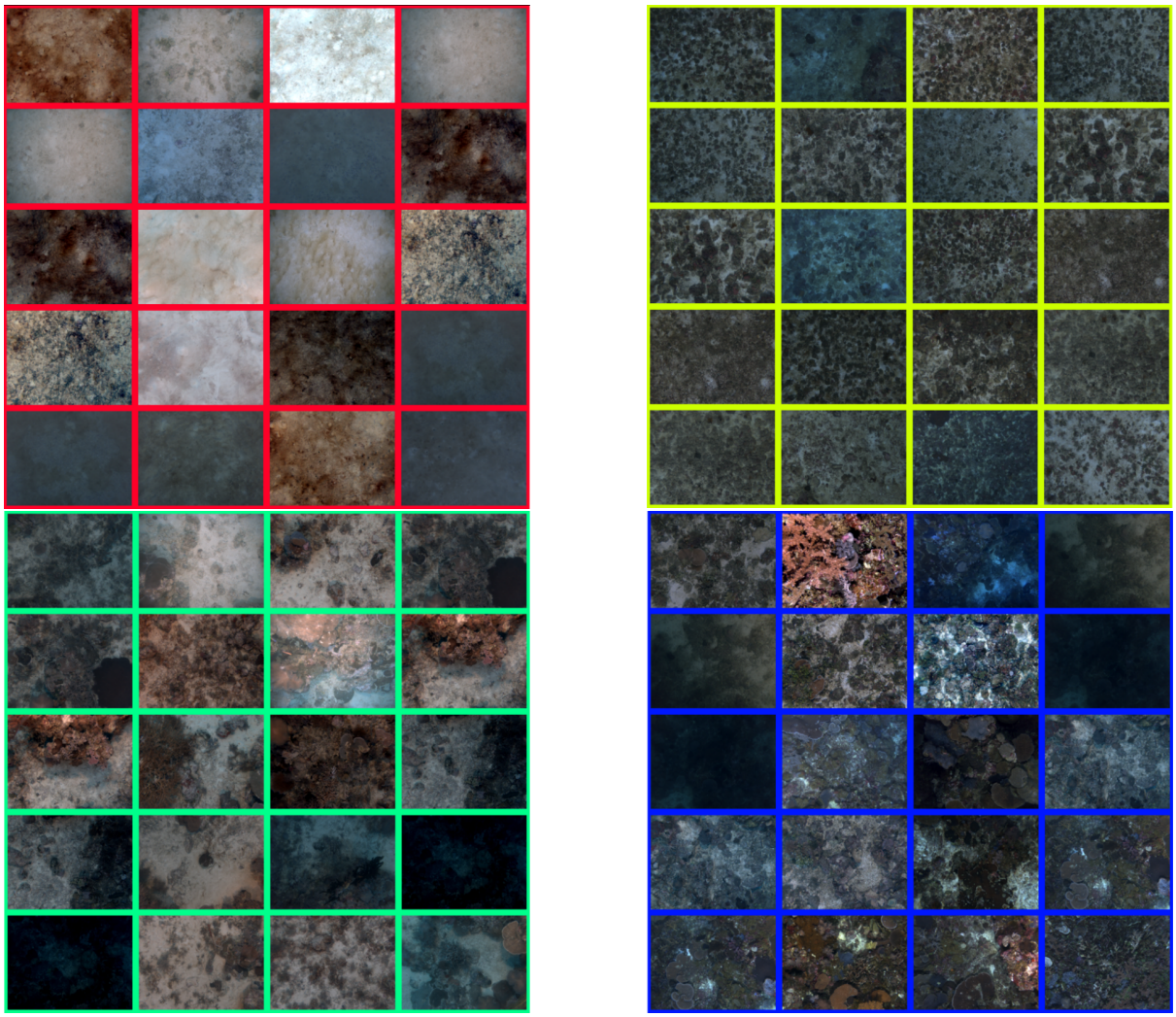


FIGURE 5.4. Sample images from each of the four simplified classes, in order from top to bottom, left to right: sand, rhodoliths,coral, halameda

it resulted in an lower average error when performing 10-fold cross-validation using Dirichlet Multinomial Regression (Section 5.5.1). It was also chosen to allow predictions to run faster, and for the Markov Chain Monte Carlo for Dirichlet Multinomial Regression later in this chapter, **significantly** reduce the number of dimensions that need to be traversed - considering the number of weights required is features $\times$ number of labels, this would correspond to the 4-label data needing $19 * 4 = 76$ vs $55 * 4 = 220$ weights, and the full 24-label case requiring $19 * 24 = 456$ vs $55 * 24 = 1320$ weights.

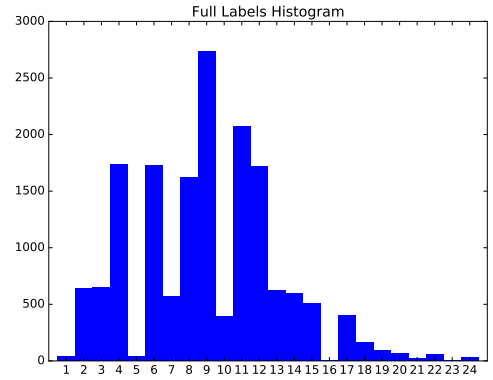


FIGURE 5.5. Distribution of labels in argmax labels

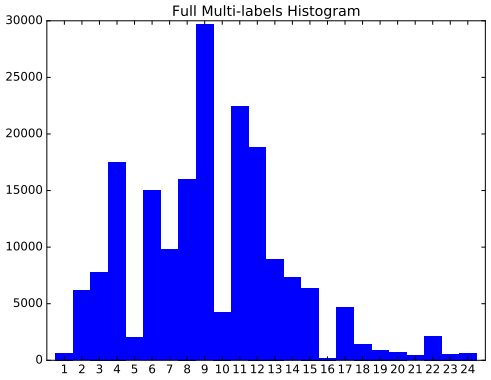


FIGURE 5.6. Distribution of labels in original multi-label outputs

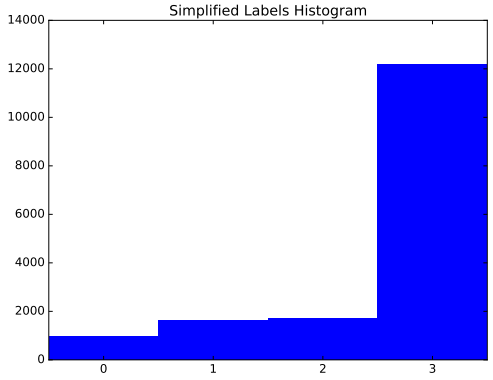


FIGURE 5.7. Distribution of simplified labels in original dataset

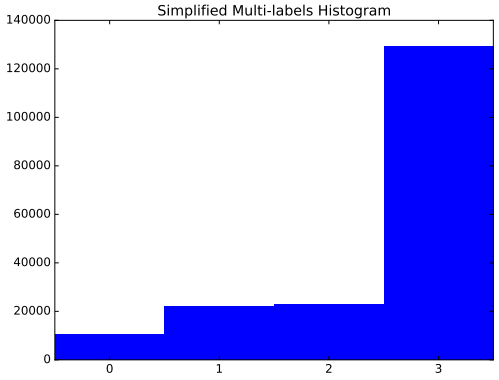


FIGURE 5.8. Distribution of simplified labels in multi-label outputs

5.3 Deterministic Approaches (Single Output)

We first briefly review the machine learning techniques more commonly used in benthic habitat mapping first, to get an idea for the sort of maps generated as well as their performance for the given dataset. To quantifiably compare their predictions, we calculate their unweighted f-scores. The *f-score* of predictions are a measure of accuracy in classification problems that takes into account both precision and recall across each possible label, and is calculated by $2 * \frac{\text{precision*recall}}{precision+recall}$. The use of unweighted f-scores means, we calculated the *f-score* separately for each label in the predictions, and simply took the average of them. This was chosen in

Algorithm	10F-CV F-score	10F-CV Accuracy	Label type
SVC	0.21514	0.75554	4 labels
LogisticRegression	0.33713	0.77001	4 labels
KNeighborsClassifier	0.4714	0.7796	4 labels
RandomForestClassifier	0.4737	0.79406	4 labels
SVC	0.10355	0.29408	24 labels
LogisticRegression	0.13335	0.31389	24 labels
KNeighborsClassifier	0.22593	0.33093	24 labels
RandomForestClassifier	0.22015	0.3405	24 labels

TABLE 5.2. Performance of common machine learning models

favour of weighted f-scores that provide a larger weight for more frequent labels as the high occurrence of sand would hide the fact that the other labels are constantly incorrectly predicted, if this was the case.

While the accuracy of the Logistic Regressor, kNN, and Random Forest Classifier are reasonable (above 0.75), the former two's f1-scores are very poor at 0.33, with the latter two at just below 0.5, which is an equally undesirable result. Looking at the ratio of available labels in the downsampled data in the 4-label case ($232, 470, 446, 3548$ for labels 0, 1, 2, 3 respectively) reveals that label 3 accounts for 0.7556 of the dataset - a value very close to the accuracy of predict. The weighted f1-score of a 'naive' classifier that always predicts label 3 has an accuracy of 0.75554 and an average f-score of 0.215 - highlighting the fact that these simpler models are not able to produce results that confidently outperform simply guessing one label for any given datapoint. Figure 5.9 visualises the predictions from Section 5.3 on the full query data for the 4 and 24-label data respectively(only showing 4-label(?) case atm! and need to include discrete label colourbar!). The Support Vector Machines (SVM) that generally provides moderately respectable real-world performance has noticably failed to predict anything other than sand throughout the query space, hinting the underlying data has complexities that require more complex models to explain it. The predictive maps generate using Logistic Regression and kNN bear noticable similarities in many areas of the map, while Random Forests identified regions that the others didn't.

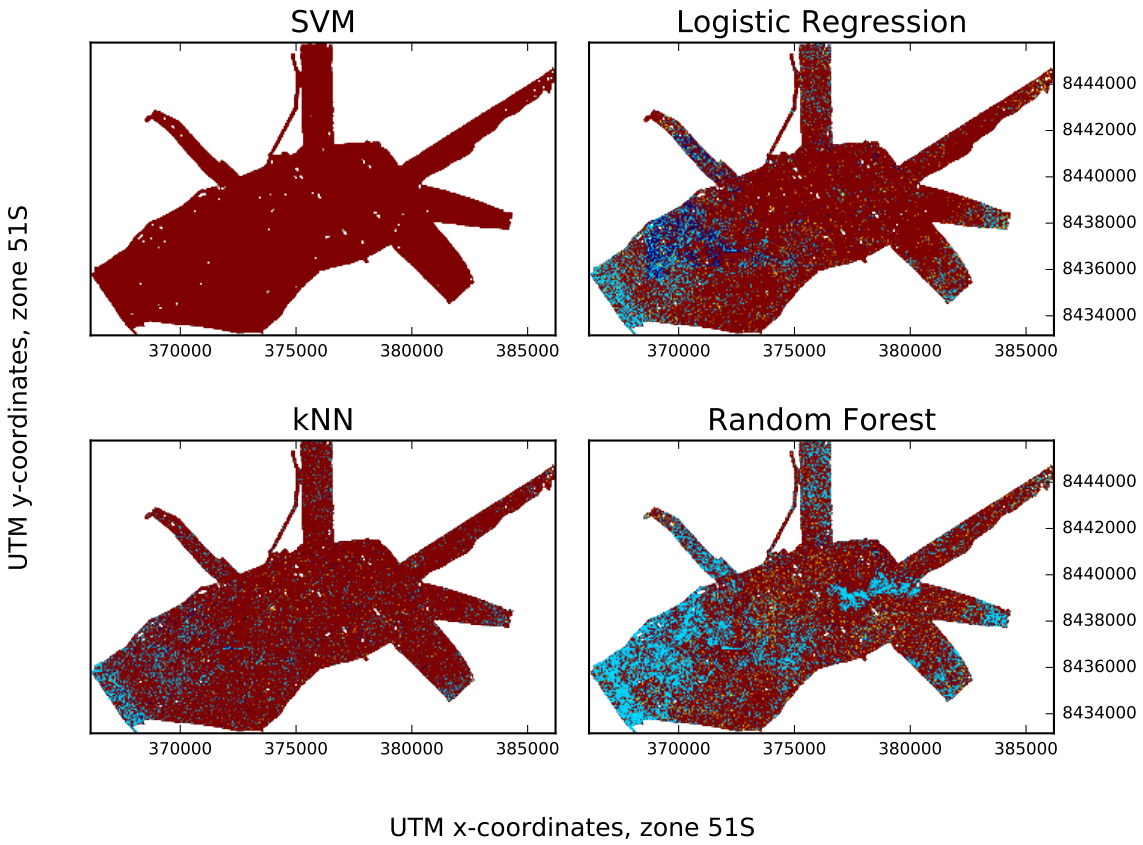


FIGURE 5.9. Full predictive map using SVMs, Logistic Regression, kNN, and Random Forests (TODO colourbar)

The maps in Figure 5.9 (with the exception of the SVM-generated one) provide some insight into where certain habitats occur in Scott Reef. However, as the results of the other three models were comparable, particularly for Random Forests and K-Nearest Neighbours, it is not obvious which one is more 'trustworthy', and what prediction to take in areas that they disagree on. One piece of information that can aid in this regard is if a level of *confidence*, which we explore in the next section.

5.4 Probabilistic Approaches (Single Output)

In this section, we will add an extra layer of information to our models' outputs - the confidence of the label predictions made. When predictive variance at each point is given, a large variance would indicate a low level of confidence as the predicted value is any within a large range, whereas a small variance indicates a high level of confidence in a prediction, as the possible range of values is only a small one. For this, we need probabilistic models that naturally provide this desired variance in its predictions. In particular, as we saw in Section 3.3, Gaussian process classification is a good option for this.

5.4.1 Gaussian Process Classification

While f-scores and accuracy are still assessed via 10-fold cross validation whilst using Gaussian process classification, we introduce another metric, area under the receiver operating curve, to make use of the fact that the one-vs-all Gaussian process classifier provides a *probability* of each label's membership at each datapoint. This encapsulates that at any given point, predictions will (almost) never be 100% certain - every single possible label, however unlikely, will have a probability assigned to it. To visualise the predictions, the most likely label at each point is taken to be the true label. In addition to the information behind these labels being probabilistic, we can use the variance to quantify the uncertainty of a particular probability. For example, given the probabilities for class membership for a particular label in the range $[0, 1]$ 0.74 ± 0.5 and 0.74 ± 0.08, both would result in an positive result (1.0), but the latter is considerably more *confident* than the former.

TODO **Area Under The Receiver Operating Curve**

AUROC	Accuracy	F-score	Labels used
0.85	0.83	0.53	4 labels
0.55	0.39	0.32	24 labels

TABLE 5.3. Gaussian proces 10-fold cross validation errors on full and simplified labels

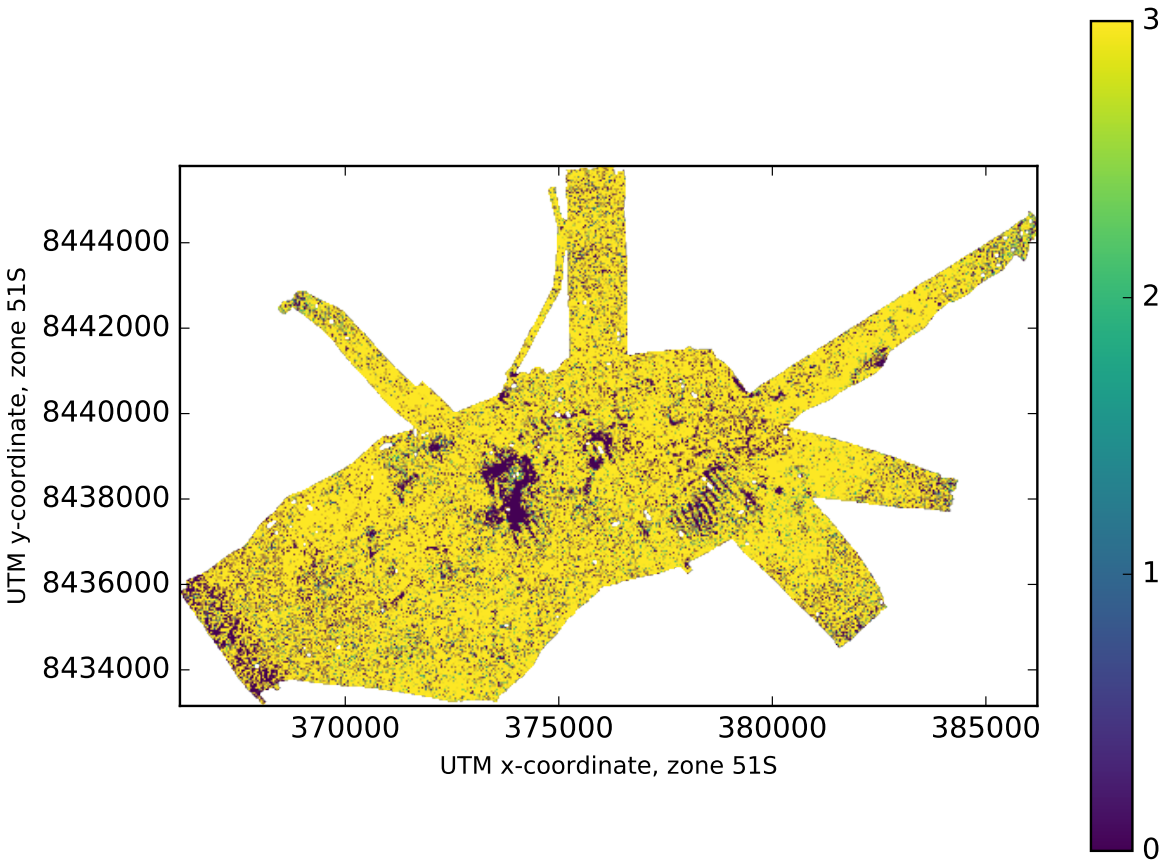


FIGURE 5.10. Gaussian process classification predictions over full query space for simplified labels

While taking the most probable per label per point gives an approximation on how the predictions over the query space look like, using it in conjunction with the original probabilities as well as the variance at these locations can provide further insight that allows the different information available in a Gaussian process to be used to their full extent. Starting with Figure 5.11, the underlying probabilities of the predictions reveals that a majority of the points being predicted only state a probability of $0.624 \sim 0.632$, and none above ≈ 0.670. This is quite revealing, as it would prevent any absolute statements from being made about any portion of the predictions over Scott Reef, but allow them to be made stating the level of confidence.

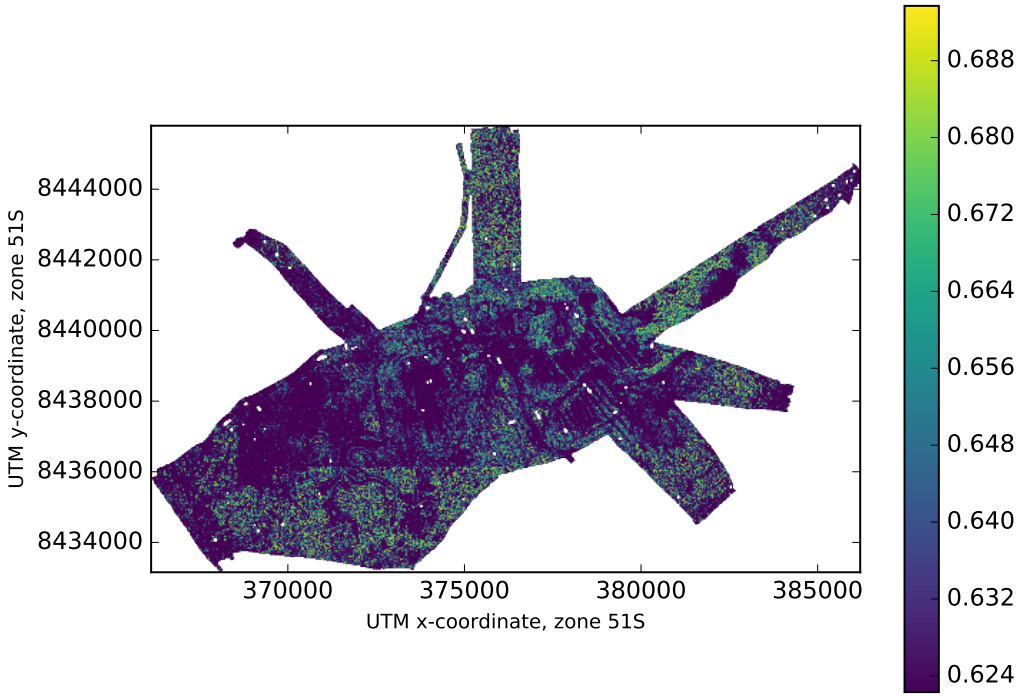


FIGURE 5.11. The largest probabilities at each point over the 4 possible labels

This highlights the advantage over the derministic methods previously explored - with only accuracy and f-scores to go by, the lack of any probabilistic information makes it difficult to make decisions when dealing engaging in tasks as important as the management of Earth's oceans.

The predictive variance of a Gaussian process can provide further information about the predictions on top of confidence. Although the observed maximum probability of predicting any one point was 0.7, this was not yet taking into account the *variance* over each of these probabilities. Figure 5.12 reveals that large swaths of the predictions vary by more than ± 0.5, with only limited, scattered areas in the $0.34 \sim 0.44$ range, and only a few isolated areas below 0.28.

Predictions on the original labels were not performed, as fitting 24 separate Gaussian processes to each of the labels for all the 4700 training datapoints, then performing predictions across all $500,000$ points required extensive computational resources that were not available for the duration of this study.

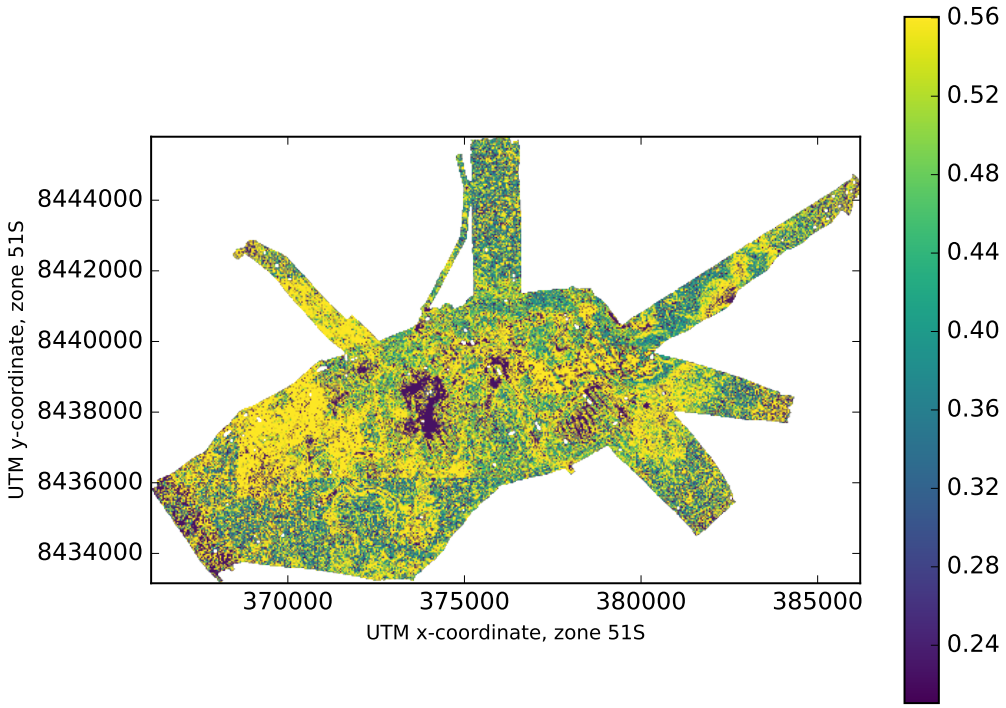


FIGURE 5.12. Standard deviations for the largest probabilities (irrespective of label) for each point in the query space

5.4.2 Ensemble Gaussian Process Approximations

Although Gaussian processes provide the benefit of the possible variance for every prediction, the matrix inversion steps required and their $O(n^3)$ complexity prevents scaling of fitting the model beyond several thousand points (that even on high-end consumer hardware is impractically time-consuming), and predictions an order of magnitude above that. Using the naive Gaussian process, model fitting on the aggregated 4-label data with 5000 data points took over a day(double check this again!), with the predictions of the $500,000$ training points taking another day. It would be helpful to note that the formulation of Gaussian process classification used in this study (binary one-vs-all classifiers per label) means the complexity not only scales with number of points, but also in the number of possible labels, with each label requiring another underlying Gaussian process to account for it - for example, data containing 24 labels would require 24 separate binary Gaussian processes. Due to hardware constraints, the 24-label

cases were not run to completion using the naive approach. As a point of reference, model fitting and predictions were timed for the basic Gaussian process, but only for a subset of the data, as shown below.

Points	Labels	Time Taken (hh:mm:ss)
1000	4	00:00:37
2000	4	00:02:12
3000	4	00:11:53
4000	4	00:33:26
4700	4	00:47:41
1000	24	00:00:54
2000	24	00:07:03
3000	24	00:23:42
4000	24	00:39:43
4700	24	01:07:15

TABLE 5.4. Gaussian process model fitting runtimes, using gradually increasing number of points for both simplified and full-label cases.[2]

Training points used	Labels	Time Taken (hh:mm:ss)
1000	4	00:01:23
2000	4	00:04:18
3000	4	00:10:02
4000	4	00:17:35
4700	4	00:32:41
1000	24	00:52:14 (s)
2000	24	
3000	24	
4000	24	
4700	24	

TABLE 5.5. Gaussian process prediction runtimes, using gradually increasing number of points for both simplified and full-label cases.

(TODO - looks like predictions are not able to even run properly for the 24-label data. might need to cut out half of above table and give a brief justification)

[2]Note that in these two tables, the time required in the 4 label cases vs the 24 ones appears to scale better than linearly. The only reason for this is the use of a 32-core processor, allowing parallelisation to hide the actual 'computing hours' involved in the computations. A more accurate representation of the time needed independent

On the other hand, the benefits of Gaussian processes need not be sacrificed on account of this disadvantage of Gaussian processes, as approximations exist to break the original dataset into smaller chunks, allowing parallelisation and model fitting per smaller set of data that is only limited by the available hardware. As seen in Section 3.4, one such method is to use ensembles of Gaussian processes that allow trivial parallelisation. These experiments were carried out on **d2.8xlarge** Elastic Container (EC2) instances from Amazon, with the following specifications:

Instance Type	vCPUs*	Memory(GB)	Physical Processor	Clock Speed
d2.8xlarge	36	244	Intel Xeon E5-2676v3	2.4Ghz

TABLE 5.6. Amazon EC2 Instance Machine Specifications

To illustrate the usefulness of these approximation methods, experiments were run to measure their accuracy and f-scores, but also the time needed to run them particularly compared to naive Gaussian processes and usefulness of the maps - providing a point of reference when looking at Dirichlet multinomials in the next chapter. The abbreviations of the ensemble methods will be used in the following tables: product of experts (PoE), generalised product of experts (GPoE), Bayesian committee machines (BCM), robust Bayesian committee machines (rBCM).

of multiprocessing would be to multiply the number of labels used by the time taken for each operation (minus some, for the overhead of inter-process communication between Python's process Pools.

Ensemble method	F-score	Accuracy	AUROC
PoEGP	0.35	0.76	0.71
GPoGPE	0.33	0.75	0.69

TABLE 5.7. Gaussian process approximation results for simplified (4) labels

Ensemble method	F-score	Accuracy	AUROC
PoEGP	0.17	0.18	0.34
GPoGPE	0.19	0.21	0.34

TABLE 5.8. Gaussian process approximation results for full 24 labels

Ensemble Type	Labels used	Time Taken (hh:mm:ss)
PoE	4	00:00:23
PoE	24	00:00:37
GPoE	4	00:00:23
GPoE	24	00:00:40

TABLE 5.9. Gaussian process ensemble training runtimes for all 5000 Training points

Ensemble Type	Labels used	Time Taken (hh:mm:ss)
PoE	4	00:03:35
PoE	24	01:54:47
GPoE	4	00:03:31
GPoE	24	01:53:30

TABLE 5.10. Gaussian process ensemble prediction runtimes for all 500000 test points

Given that random forests were able to best fit the training data in Section 5.3, it is promising that the the simplified generalised product of Gaussian process expert predictions, whereby the label with the maximum probability at each point is taken to be the 'absolute' truth, bears close a resemblance to it. If Gaussian processes were used in practice, there is still the variance that can be used to make better use of the predictions.

While the product of experts and its generalised counterpart appear to have the comparable performance measured on error alone, the former failed to predict labels $0, 1, 2$ in the simple case, and the corresponding labels for the full 24. The generalised product of experts however, performed quite well, identifying similar habitat regions to the Dirichlet multinomial, as shown in Section 5.5. Even with these improvements to running time whilst retaining probabilistic information, there is still important functionality missing from these models - the counts of other labels at every point that was available in the original data. The methods explored thus far are not able to deal with such multi-output data, and other models must be used to do this. Moreover, the level of parallelisation used to speed up operations would only be available on server hardware that can become expensive if constantly relied upon. Although scaling

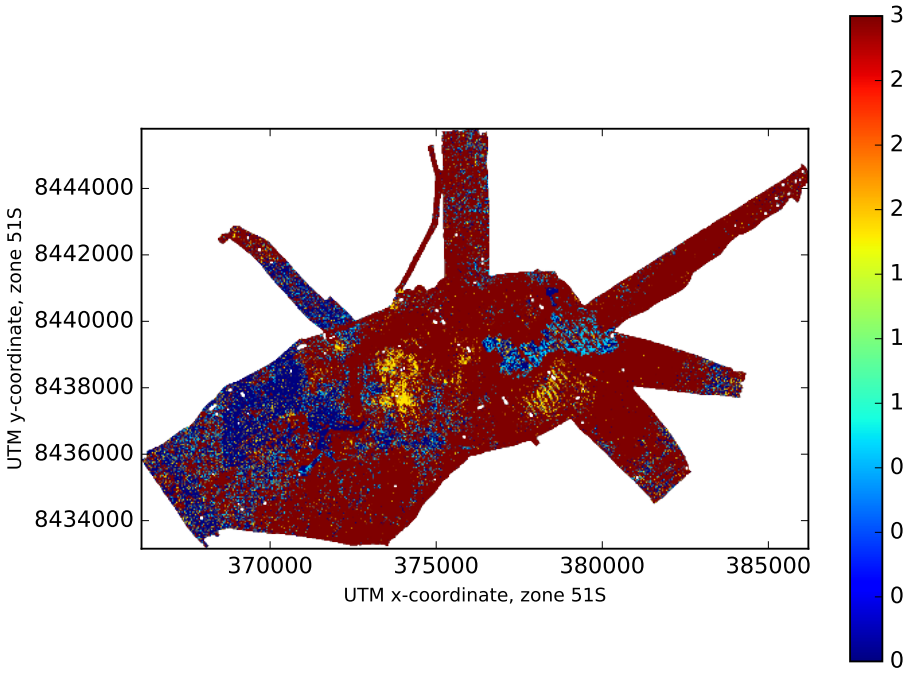


FIGURE 5.13. Argmax map of generalised product of Gaussian process experts for simplified labels

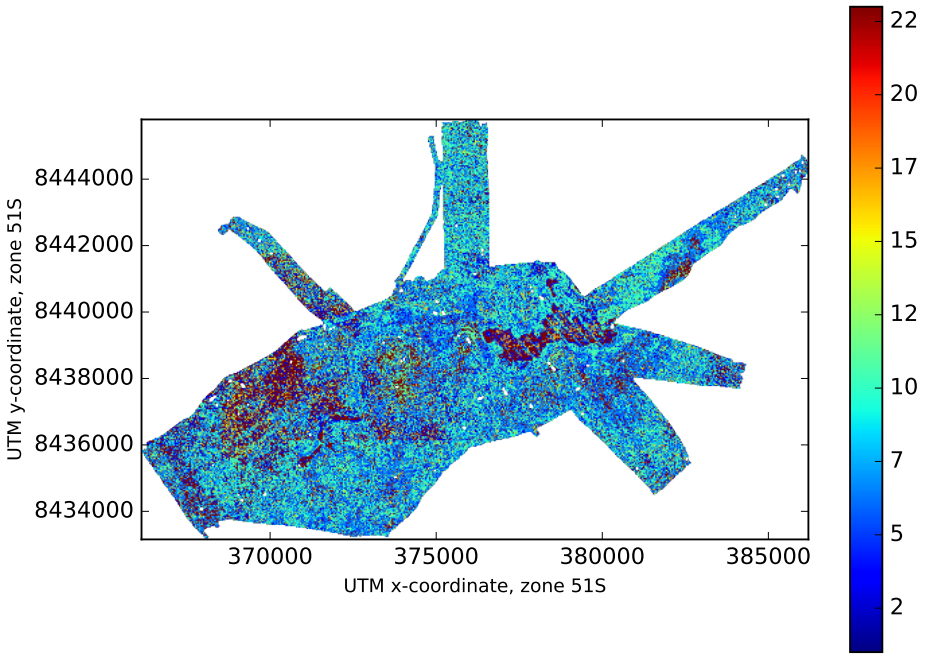


FIGURE 5.14. Argmax map of generalised product of Gaussian process experts for full 23 labels todo: match colours better, fix colourbars

FIGURE 5.15. herp

Gaussian processes was possible here using hardware and approximation methods, sacrifices in correctness were made.

To use in live scenarios, such as an AUV with previous bathymetry data collecting new images, the time it would take to run approximations before being able to calculate a desired path would render the vehicle inactive for most of the time. The several minutes achieved by the ensemble methods on the simplified data above was a result of parallelising over 30 cores and being able to utilise over 200GB - not the sort of hardware that would be appropriate for a small deep sea vessel. Not to mention, the sort of post-processing required in simplifying labels requires human intervention, and would not be something that can be automated on an expedition into previously unexplored areas. Hence, different methods would be needed that can perform similar tasks that can provide measures of uncertainty in realtime, all while being able to run quickly on hardware with limited capabilities.

5.5 Multi-Output Predictions

Looking at the deterministic maps from Figure 5.9 as well as Figure 5.13 and Figure 5.14 using the simplified ensemble Gaussian process approximations, it can be observed where clusters of certain habitats are - but what can't be easily obtained, or at least automated without non-trivial extra effort, is identifying exactly *where* these clusters are, and the frequency of co-occurrence between the different habitats. This is a consequence of only having a single label per point, but considering the area covered by a single data point, it is unrealistic to imagine the entire surface being only a *single* label. Thus, we explore how to predict the **distribution** of labels at each point as represented in the original data, to provide richer habitat maps that naturally illustrate the co-occurrence of different habitats.

As a means of effectively visualising the separate labels, we need to look at the normalised distribution of habitat classes for each label separately. In the maps below created from each model's respective predictions, each class is represented on a separate heatmap, with the occurence (with a maximum of 1, when an area is predicted to *only* contain that label) indicated by the colour bars included above each map. This allows initial observations to be made of

Labels used	Root Mean Squared Error
4	0.17664
24	0.05916

TABLE 5.11. Dirichlet Multinomial Regression average error for the two label sets

where certain labels are more abundant than others. This representation allows a user/viewer to easily manually identify where and which labels have a high occurrence (without being required to constantly check which specific colour a label was, etc.), but also larger areas where habitats co-occur.

5.5.1 Dirichlet Multinomial Regression

The last model used was the Dirichlet Multinomial, which incorporates the constraint where predictions over any number of labels had to sum back to 1, as a result of the Dirichlet distribution component. This means that from a mathematical standpoint, these predictions will be more 'correct' for multi-output labels than all the previosly explored models - but we also want to see how they hold up in practice.

To assess initial performance, the weights and hence the α parameter was obtained via the maximum a posteriori estimation.

For the simplified labels, we can see some similarities with the models generated using random forests and Gaussian processes (Figure 5.9, Figure 5.13), with certain regions matching up to different predictions. For example, all predictions were able to agree on the general dominance of label 3 throughout the reef, but due to the single-output nature of the previous methods, this dominance also meant it was the *only* label to appear in a majority of the predictive maps. Using this model, the actual occupancy rate of label 3 can be determined, instead of only being able to see 'all sand'.The ability to quantify the presence of certain habitats also becomes quite trivial, as a single line of Python code can provide information such as label 3 occurring at a rate of more than 0.5 in 73% of the predicted query space, whereas labels 0, 1, and 2 only occur at a similar rate 4.7%, 9.9%, and 2.3% of the time.

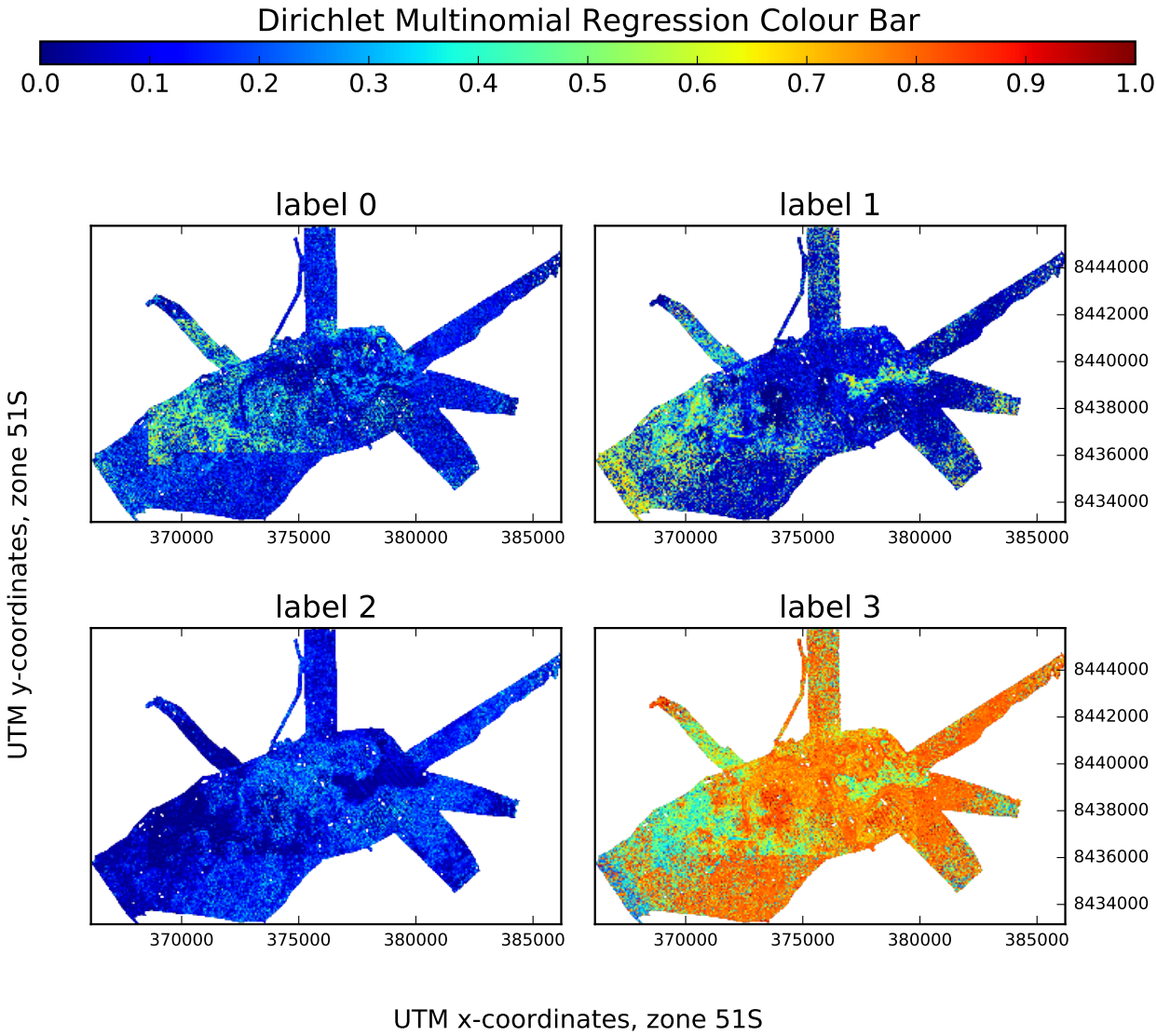


FIGURE 5.16. Distribution heatmaps over each label (in the simple 4-label case) for Dirichlet Multinomial Regressor output on query points

By observing the entropy of the Dirichlet distribution using the α parameter (cite equation), observations about the confidence of predictions in different regions can be made. Low entropy areas indicate a low variance in the data and hence a confidence in the predictions made in that area, with the reverse being true for high-entropy areas. Figure 5.17 shows some key regions (in dark blue) that are very low entropy, with large regions of moderate entropy (cyan) predictions spread out over the query space. Of note is the fact that the noticable areas of co-habitation between labels 1 and 3 where they are close to an even $50 : 50$ split are very low entropy,

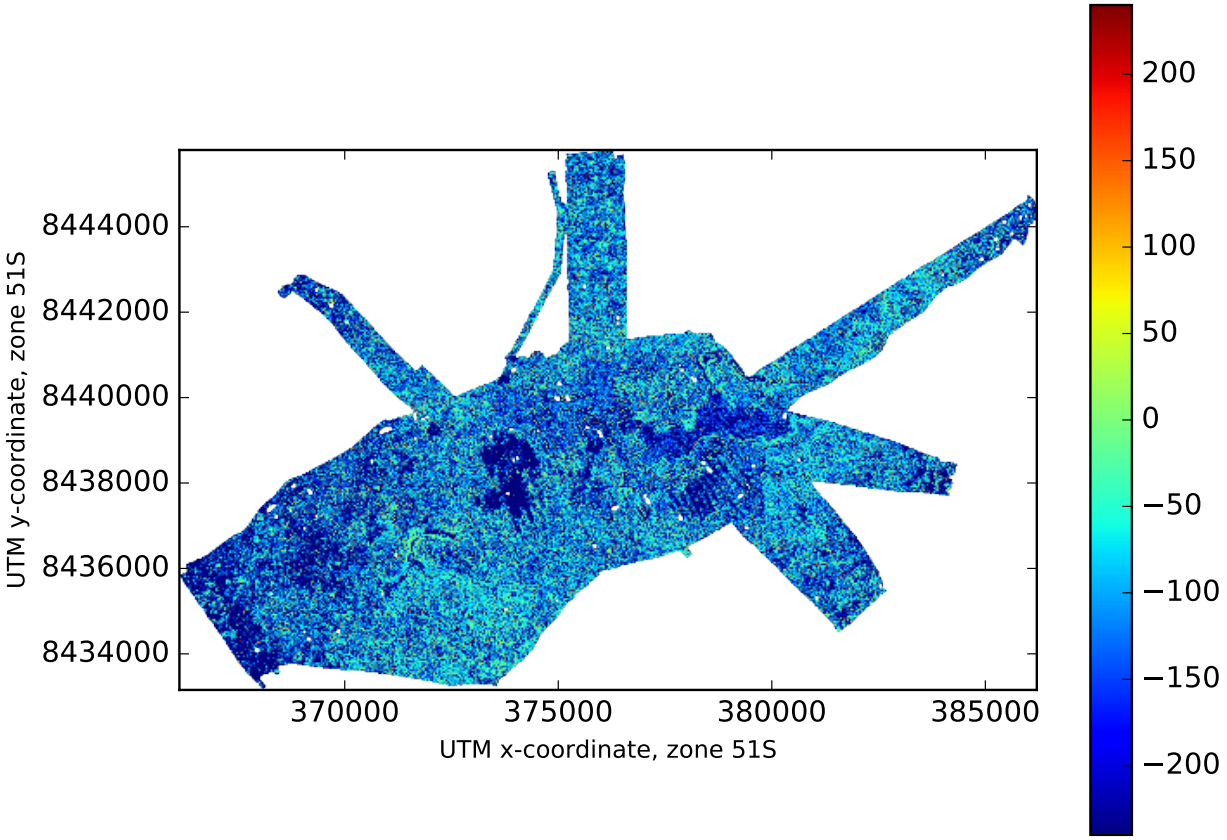


FIGURE 5.17. Entropy plot over all query points for the simplified labels.

highlighting the Dirichlet multinomial's ability to detect when a subset of labels within multi-output data correlate with one another with confidence.

In Figure 5.17 the dark-blue low-entropy region on the right hand side of the map, as well as the large area on the left, contain comparatively denser low entropy regions than the remaining points. Referring back to the per-label predictive distributions, these are also areas with noticable cohabitation between labels $0, 1, 3$. What becomes apparent is that in the areas where the DM is confident of a mix of certain set of predominant labels, compared to the GP from earlier that is instead equally uncertain of each of them with a considerably higher variance, which is misleading information when taken at face value. For example, this sort of uncertainty may be

taken into consideration purposes, where autonomous vehicles are used to collect data, or in making decisions with regards to conservation efforts. In the first scenario, resources are being wasted on areas where models such as the DM can be confident of a particular distribution of labels, whereas in the second, important conservation actions may be withheld if the *certainty* of information is brought into question. For example, in an area that contains a particular mix of coral and bleached coral, a DM has the potential to make a confident prediction of their coexistence, whereas a GP would make predictions where their respective probabilities in a one-vs-all classifier may be close to their distribution in the area, but have a high noise factor.

5.5.2 Dirichlet Multinomial Predictive Map Variance

As the above results from Dirichlet multinomial regressoin above were obtained using the Maximum a Posteriori (MAP) estimate of the parameters underlying the Dirichlet distributions α values, only the single set of optimal parameters were used, with none of those within the rest of the posterior distribution tested. To confirm that the maps generated via optimisation using MAP, Markov Chain Monte Carlo (MCMC) was used to obtain draws of weights from the posterior distribution. The purpose of this was to be able to obtain chains that had reasonably converged, and then observe whether the habitat maps created using all these weights would generally agree on the presence and distribution of habitats.

To calculate convergence using the Gelman-Rubin r-hat statistic, 3 MCMC chains were run simultaneously for both the simplified (4) labels and the full set of labels (24). The former required $4 * 19 = 76$ weights (number of labels $\times$ number of features), and the latter $24 * 19 = 456$ weights. Due to the large number of dimensions being explored by the MCMC, the 3 chains in both cases were not able to fully converge to 1.0. The weights for the simplified labels were close to convergence, with an r-hat score of 1.065, although 4 of the 76 weights were above 1.2 $(1.45, 3.53, 1.43, 1.21)$. The weights for the full 24 labels fared considerably worse, with an r-hat score of 1.429, with only 243 of the 456 weights being below 1.1.

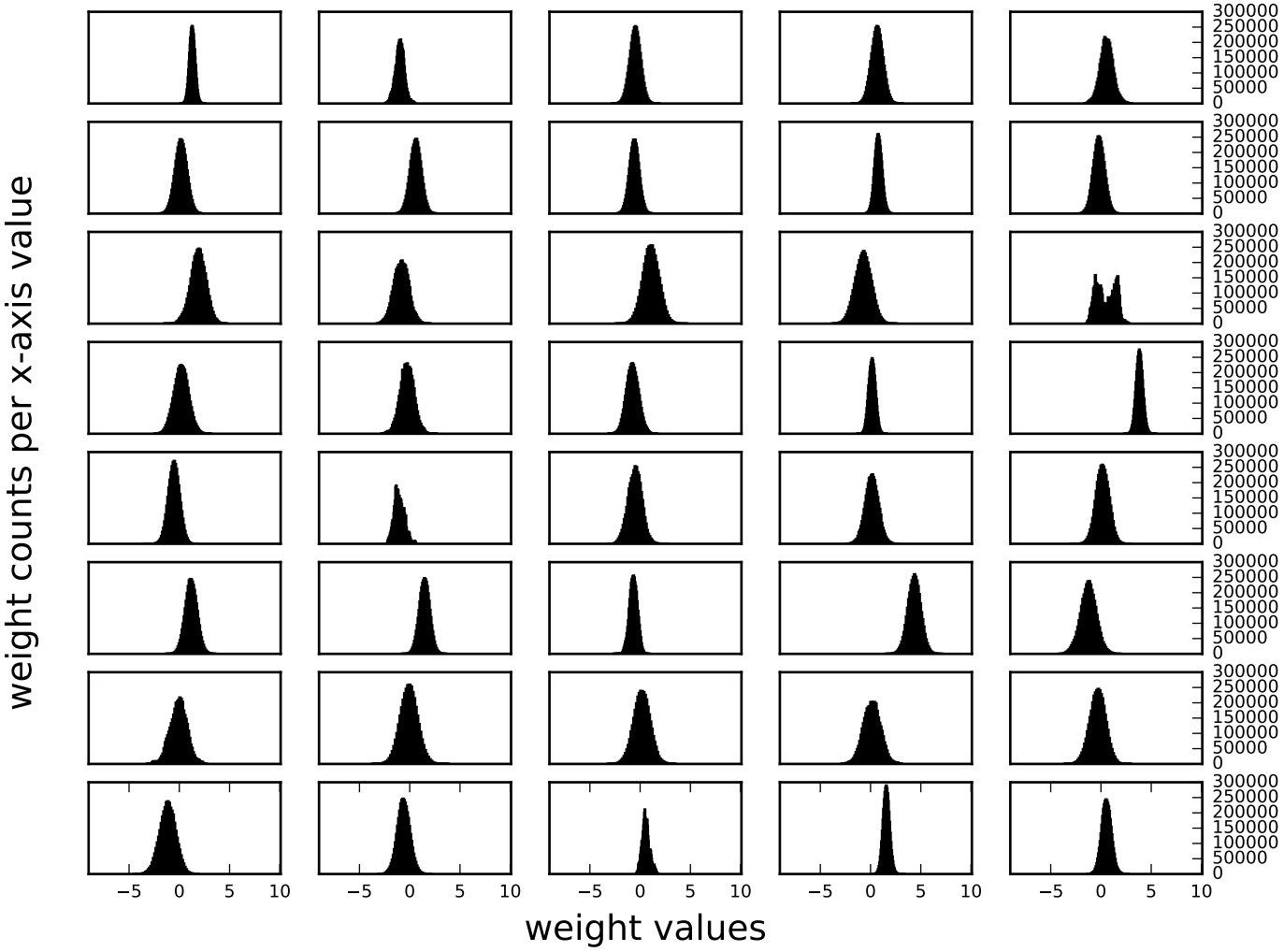


FIGURE 5.18. MCMC weights for 4-label, 19-dimension data case (weights $1-40$ of 76)

With the exception of the 15-th weight that has two peaks, it appears that the likelihood of all others is aggregated all at specific value with moderately sharp peaks. Considering the less standard likelihood of the 15-th weight, a comparison with other chains reveals that this one in particular did not converge in terms of the r-hat statistic.

As the 24-label case has over 400 weights, with a little over half converging, the 456 separate distributions have not been included here, but can be found in the appendix (not included yet. not sure if worth including in appendix though actually).

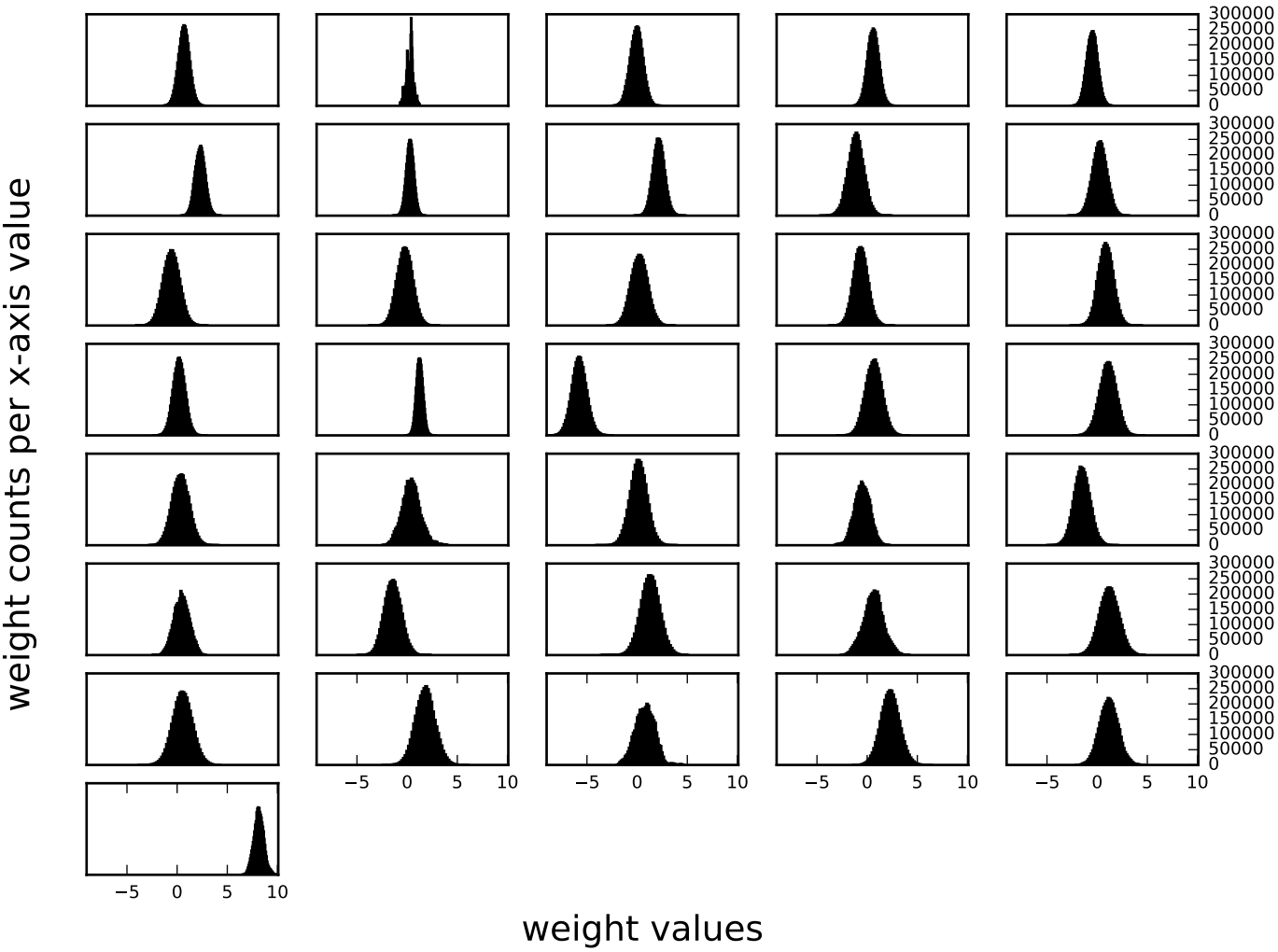


FIGURE 5.19. MCMC weights for 4-label, 19-dimension data case (weights $41 - 76$ of 76)

Seeing as the r-hat scores suggest that the weights of the Dirichlet multinomial have converged for the simplified label case, observing the maps that can be generated by every one of the draws of weights can similarly provide visual confirmation that the predictive maps are mostly

consistent. To condense the information contained in the K dimensions of predictions, the most likely label is instead taken to be a 'single' label, similar to how the probabilities of a GP per label are 'squashed' down to one value.

5.6 Biodiversity

Another beneficial aspect of Dirichlet Multinomial Regression is that it inherently provides information about the distribution of different habitats in a given region, allowing observations on biodiversity to be made without extra steps such as clustering, which can be prohibitively expensive on datasets with millions of datapoints and tens (or more) of dimensions. Locating the co-existence of certain species would involve searching over the space of predictions for the desired distribution of habitats.

As the 4-label case already aggregated similar classes from 24 down to 4, there was minimial biodiversity to observe over the query space, requiring us to perform predictions over the full 24 labels to be able to find more abundant occurrences of biodiversity. To give a qualitative visual representation of the distribution of labels over the query space when using all 24 labels, the predictive heatmaps for each of them is shown in Figure 5.20, Figure 5.21, and Figure 5.22. The numbers in brackets next to each title indicate the simplified label they correspond to.

Given the multi-label distribution predictions, there are any number of ways to either quantatively or qualitatively measure biodiversity. As an instance of the former, strict numerical conditions are used to define biodiversity for the example below. Depending on the aim of a particular environmental study or survey though, the definition of biodiversity is entirely flexible - it may refer to the co-existence of a specific few species or habitats ignoring all others, or it may be general biodiversity that defines cohabitation between any habitats. We take the latter of these formulations to show the power provided by the Dirichlet multinomial. For any given point (and by extension, region, where this pattern occurs frequently enough that the number of points fitting the criteria are dense enough to described as a cluster), n number of labels are said to co-occur if for a given data point, the relevant area of benthos corresponding to a coordinate contains more than n labels (number of 'co-existing' habitats) that occur at a rate of at least e, where $0 < e \leq 1$. Given this specific 'version' of biodiversity, e would naturally need to decrease as n increases - while it is possible to have large areas where at least two labels occur at a higher rate than 0.15, this is of course not possible for seven points, for example $(7 \times 0.15 = 1.05)$.

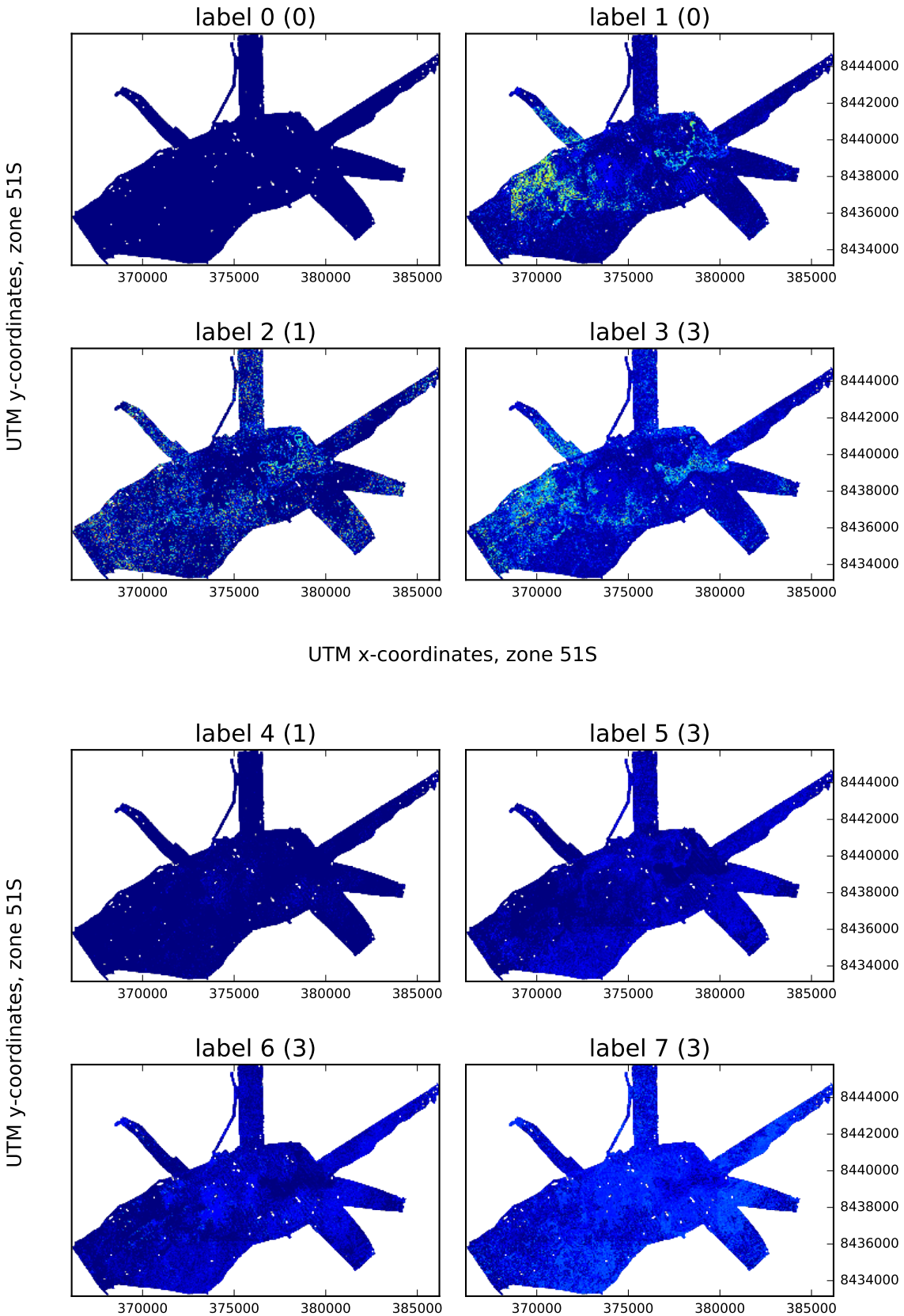
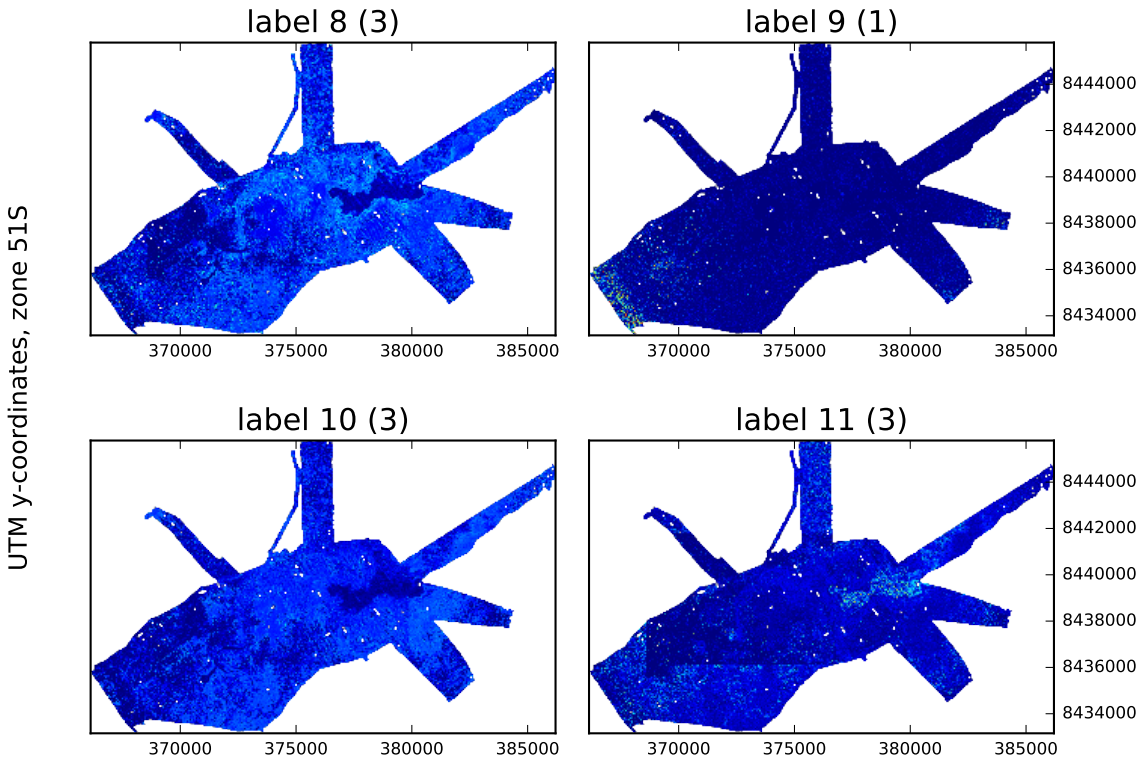


FIGURE 5.20. (need to redo these to show more colour @_@)



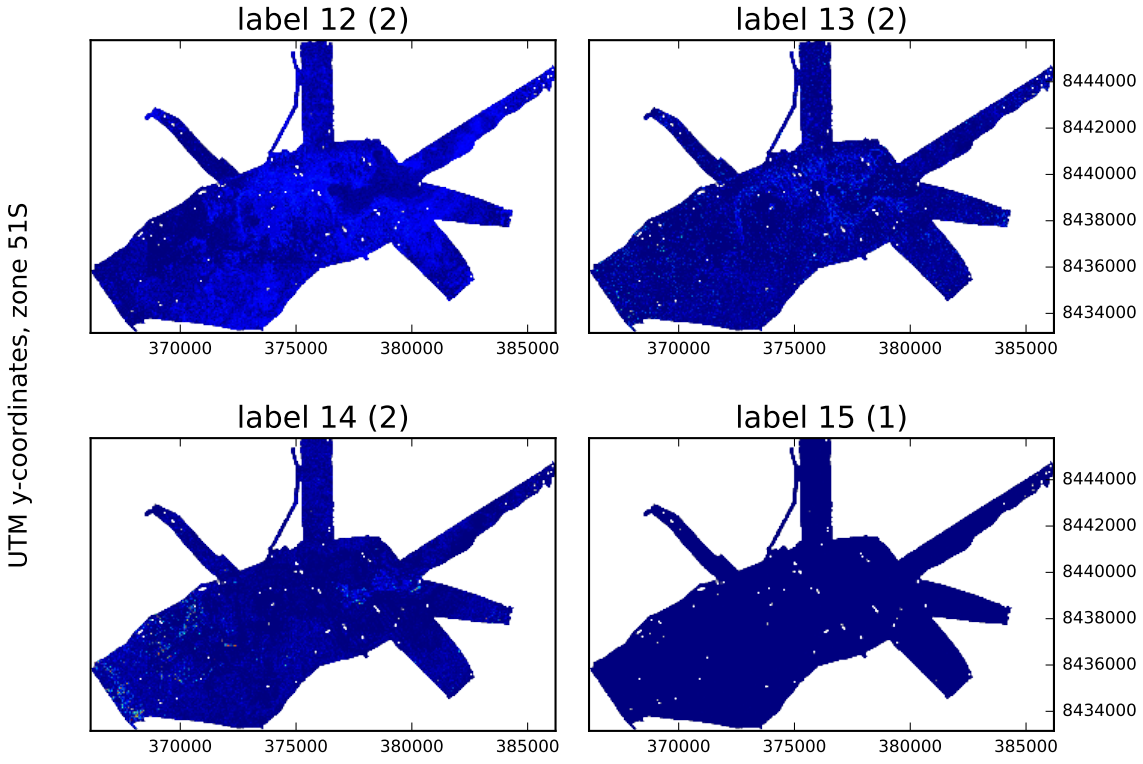


FIGURE 5.21. (need to redo these to show more colour @_@)

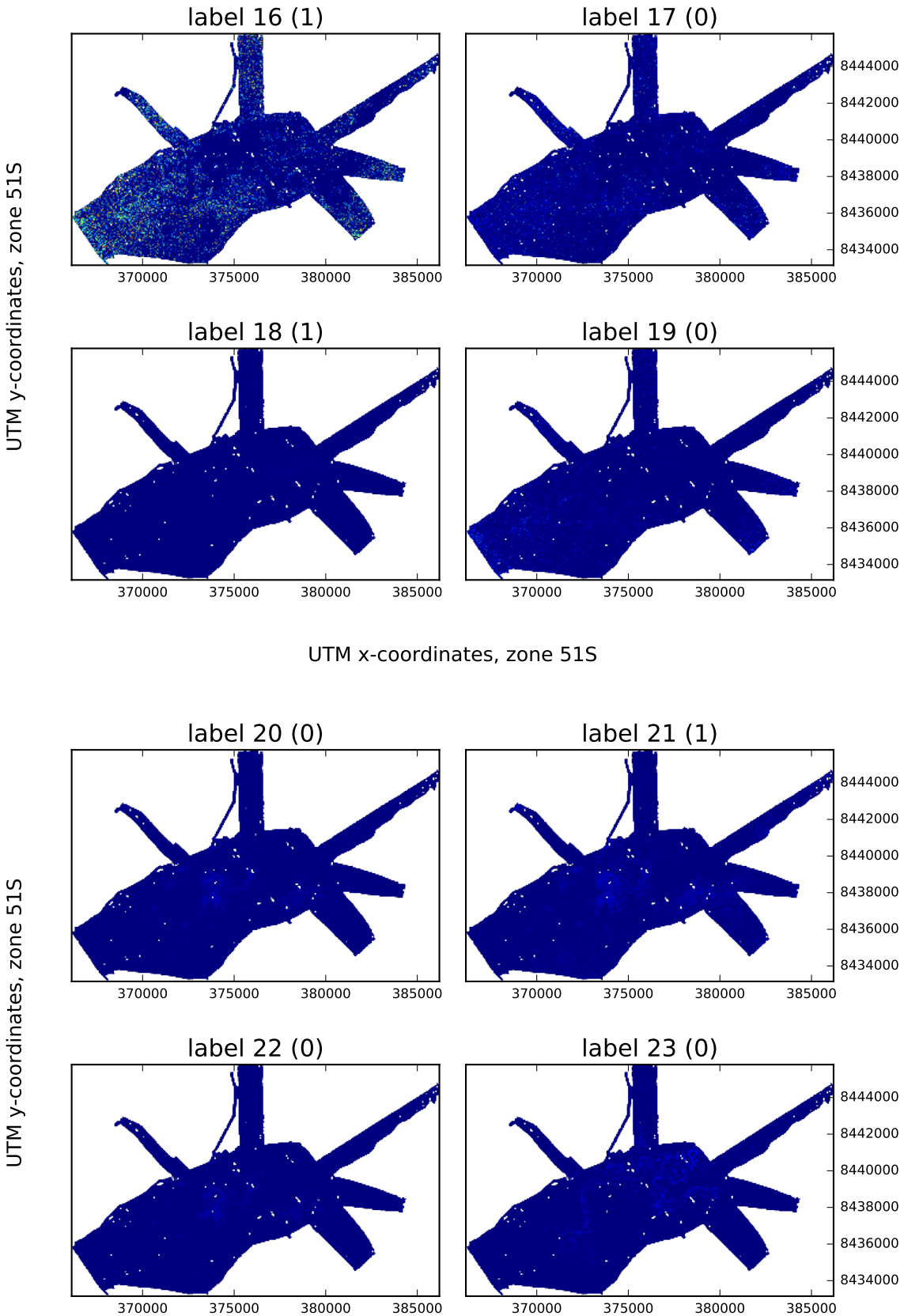


FIGURE 5.22. (need to redo these to show more colour @_@)

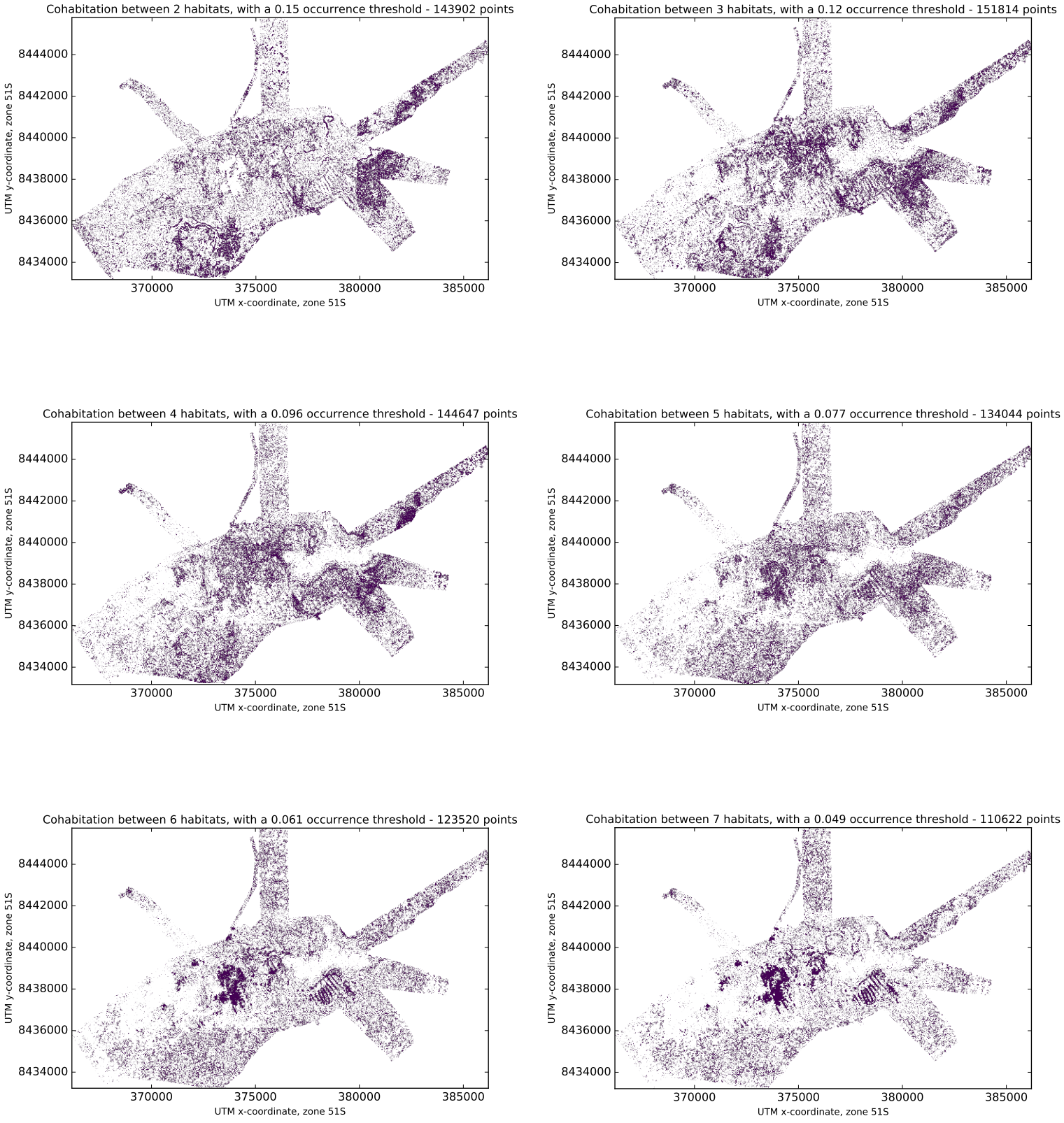


FIGURE 5.23. Cohabitation for the stated occupancy threshold for the label of each plot, from $2-7$.

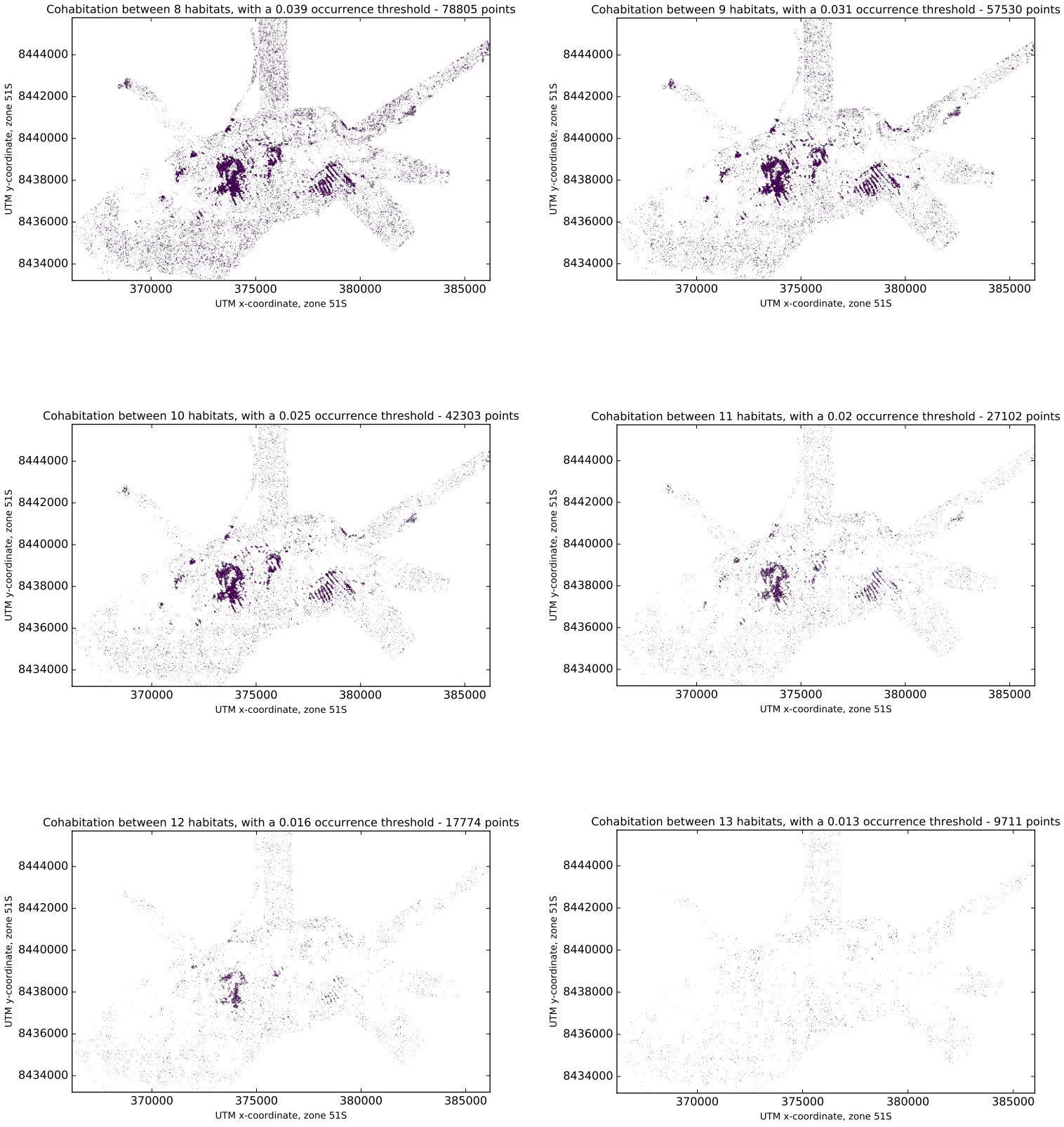


FIGURE 5.24. Cohabitation for the stated occupancy threshold for the label of each plot, from $8 - 13$.

For the definition of biodiversity taken in the above plots, it is easy to observe the biodiversity over the 24 labels in general, with a general trend that biodiveristy between an increasing number of habitats, even with the decreasing threshold, results in a lower density of biodiverse areas. The initial threshold when considering 2 labels was 0.15, dropping by 10% (or 0.9 of the previous value) for each additional label considered. For these particular parameters, any signs of visibly continuous areas of biodiversity disappear when considering more than 13 labels.

This approach can be easily modified to target the tracking of specific habitats - for example, if data is collected periodically in any area where coral bleaching is suspected to occur, predictive maps can be generated at the same intervals, allowing the *changes* in biodiversity to be observed. Although such a task can be more generally achieved even using deterministic methods by simply taking a count of labels over the query space to see positive/negative trends, this is quite coarse in comparison, and specifying with accuracy the regions where change occurred involves extra work - for example, to assess if an area that was previously unbleached coral had become bleached through automated means, an individual working with the data may have to define an algorithm that checks the area around points up to a certain distance, then observe any potential changes in predictions generated from newly collected data, requiring considerably more effort in designing a method that allows analysis to be done efficiently. In contrast - performing the checks for general biodiversity using the above biodiversity across all 24 labels only took 11 seconds, with checks for individual labels take less than 1 second.

CHAPTER 6

Evaluation and Discussion

6.0.1 Limitations

- data simply not varied enough/uninteresting habitat spread in Scott Reef?
- training data doesn't explore any particular area exhaustively - hard to verify how accurate any model is even if cross validation scores are high
- from the full 24 clusters, it's apparent that some were clustered as a result of lighting, unfortunately not a desired behaviour ==> possible future work is to first 'normalize' the contrast/visual properties of the images beforehand (Get a citation for this, I think it was an ACFR paper)
- as a result of non-normalised images and hence somewhat flawed classifications, label 0 fails to be predicted often across most models tested, exacerbated in the 24-label case
  - suggests that data may be insufficient, or that certain data from images may need to be incorporated into training data

CHAPTER 7

Conclusion

Benthic habitat mapping is a relatively old concept, dating back at at least several decades when photos and videos became a viable method of capturing information about the benthos (Gibson et al., 2007) as an alternative to earlier destructive methods of sediment sampling. However, the different sources of data required and the machine learning techniques needed to model the data to be able to predict properties about the benthos did not become more readily available until relatively recently. With tools such as multibeam echosounders for collecting acoustic backscatter data at scale, as well as extensive visual imaging of the benthos via autonomous underwater vehicles, it has become possible to collect large amounts of data about Earth's ocean to work with to gain a deeper understanding. Many studies to date have used machine learning algorithms such as SVMs and random forests on different types of data in different habitats and shown moderately good results, while more state of the art methods employ probabilistic methods such as Gaussian processes (Bender et al., 2012) that capture a richer set of information in the form of both predictions over possible labels as well as certainties around these predictions, or Gaussian mixture models (Ahsan et al., 2011) that are able to identify clusters that are individually multivariate Gaussians.

In this study, application of products of experts models was applied to Gaussian processes to lift the usual limitation on data sizes of several thousand points due to the time that would be needed to fit them. In fact, due to the requirement of multiple Gaussians processes to form GP classifiers, this several thousand is effectively scaled down by a factor proportional to the number of classes in the data. Experiments showed that for aggregated labels, the ensembles of experts were comparable in performance to the full GP, though the performance gap grew when dealing with the 24-label case, but keeping in mind that the GP became unusable on the

full query data for predictions due to the time required. The predictions on the simple labels of the approximations also matched that of random forests and the distributions of the Diriclet multinomial's predictions.

As all the studies performed in the area of benthic habitat mapping deal with single-label outputs, they would be unable to account for and fully utilise multi-label count data, and resort to approximations by taking the most frequently occuring label per output. To assess the viability of working with multi-output data in benthic habitat mapping, the Dirichlet multinomial was used, serving exactly this purpose.

7.1 Future Work

- perform similar experiments on incrementally changing data every few years - observe biodiversity/habitat changes
- replace the simple activation function in the dirichlet multinomial with a more complex model like a GP
- previous work has been done for finding least certain areas of a GP to decide where to send AUV's to maximise resulting confidence in habitat labels - use entropy to be able to do the same with dirichlet multinomials, whilst overcoming the problem of areas with consistent heterogenous labels that otherwise confuse GPs
- combine habitat data with actual fauna distributions as well
- explore other multinomial prior distributions other than Dirichlet, such as Logit-normal distributions?

There a number of areas that would be pertinent to explore as an extension of this study, to further the usefulness of the data provided in terms of both the complexities of the underlying models used to predict data, as well as the contexts in which they are used.

Bibliography

Nasir Ahsan, Stefan B. Williams, and Oscar Pizarro. 2011. Robust broad-scale benthic habitat mapping when training data is scarce.

Asher Bender, Stefan B., Williams, and Oscar Pizarro. 2012. Classification with probabilistic targets.

C.E. Bond, A.D. Gibbs, Z.K. Shipton, and S. Jones. 2007. What do you think this is? conceptual uncertainty in geoscience interpretation. *GSA today*, 17.

Craig Brown, Stephen J Smith, and Peter Lawton. 2011. Benthic habitat mapping: A review of progress towards improved understanding of the spatial ecology of the seafloor using acoustic techniques. *Estuarine, Coastal and Shelf Science*, 92.

J. Calvert, J.A. Strong, C. McGonigle, and R. Quinn. 2015. An evaluation of supervised and unsupervised classification techniques for marine benthic habitat mapping using multibeam echosounder data. *ICES Journal of Marine Science: Journal du Conseil*, 72:1498–1513.

Rich Caruana and Alexandru Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International conference on Machine learning*. Association for Computer Machinery.

Marc Peter Deisenroth. 2015. Distributed gaussian processes. *International Conference on Machine Learning*, 2:5.

Daniel C Dunn and Patrick N Halpin. 2009. Rugosity-based regional modeling of hard-bottom habitat. *Marine Ecology Press Series*, 377:1–11.

Australian Centre for Field Robotics (ACFR). 2016. Squidle projects. `http://squidle.acfr.usyd.edu.au/viewproject#map`.

Ariell Friedman, Daniel Steinberg, Oscar Pizarro, and Stefan Williams. 2011. Active learning using a variational dirichlet process model for pre-clustering and classifiation of underwater stereo imagery. *International Conference on Intelligent Robots and Systems*, pages 1533–1539.

Robin N Gibson, RJA Atkinson, and John DM Gordon. 2007. An annual review. *Oceanography and marine biology*, 47.

Rozaimi Che Hasan, Daniel Ierodiaconou, Laurie Laurenson, and Alexandre Schimel. 2014. Integrating multibeam backscatter angular response, mosaic and bathymetry data for benthic habitat mapping. *PLoS ONE*, 9.

Vladimir E. Kostylev, Brian J. Todd, Gordon B. J. Fader, R. C. Courtney, Gordon D. M. Cameron, and Richard A. Pickrill. 2001. Benthic habitat mapping on the scotian shelf based on multibeam bathymetry, surficial geology and sea floor photographs. *Marine Ecology Progress Series*, 219:121–137.

Vanessa Lucieera, Nicole A. Hilla, Neville S. Barretta, and Scott Nichol. 2013. Do marine substrates âĂŸlookâĂŹ and âĂŸsoundâĂŹ the same? supervised classification of multibeam acoustic data using autonomous underwater vehicle images. *Estuarine, Coastal and Shelf Science*, 117:94–106.

Aaron Micallef, Timothy P. Le Bas, Veerle A.I. Huvenne, Philippe Blondel, Veit Huhnerbach, and Alan Deidun. 2012. A multi-method approach for benthic habitat mapping of shallow costal areas with high-resolution multibeam data. *Continental Shelf Research*, pages 14–26.

National Aeronautics and Space Administration(NASA). 1996. Display photos database record - sts080-734-20.

OzCoasts. 2015. Benthic habitat mapping: Mapping overview. `http://www.ozcoasts.gov.au/geom_geol/toolkit/mapoverview.jsp`.

Oscar Pizarro, Stefan B. Williams, and Jamie Colquhoun. 2009. Topic-based habitat classification using visual data. *OCEANS 2009 - EUROPE*.

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. Gaussian processes for machine learning.

Jan Seiler, Ariell Friedman, Daniel Steinberg, Neville Barrett, Alan Williams, and Neil J. Holbrook. 2012. Image-based continental shelf habitat mapping using novel automated data extraction techniques. *Continental Shelf Research*, 45:87–97.

D. Steinberg, A. Friedman, O. Pizarro, Williams, and S.B. 2011. A bayesian nonparametric approach to clustering data from underwater robotic surveys. International Symposium on Robotics Research.

Nurhalis Wahidin, Vincentius P. Siregar, Bisman Nababan, Indra Jaya, and Sam Wouthuyzen. 2015. Object-based image analysis for coral reef benthic habitat mapping with several classification algorithms. *Procedia Environmental Sciences*, 24:222–227.

addtocontentstoc

APPENDIX A

Appendix

---

things