Multi-output and Probabilistic Large Scale Benthic Habitat Mapping

JUSTIN TING

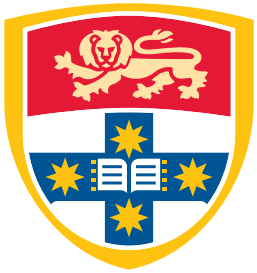
SID: 430203826

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

25 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. A considerable portion of work done in the area uses deterministic methods that strictly assign only one label to a given bathymetry data point, while more advanced models provide probabilistic results over all possible labels at any one point, also similarly only representing a single output. However, like the majority of real life classification problems (citation here perhaps), habitat mapping is intrinsically a multi-label problem for any data collected at a resolution low enough to be economically feasible to be performed at a large scale. In this paper, we explore advantages of having probabilistic class outputs as well as treating benthic habitat mapping as a multi-output problem, particularly when working with relatively low resolution bathymetry data, compared to the primary method of deterministic, single-output methods explored in existing literature.

Acknowledgements

The thanks go in here.

Contents

List of Figures

List of Tables

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 management, preseveration, etc. purposes. The process used is called benthic habitat mapping, which is the process of generating predictive maps of different habitat types at the bottom of a body of water. Most studies looking to create benthic habitat maps share some basic key steps - acoustic data is used to estimate properties about the surface of the water, which are then mapped to, using machine learning algorithms, *in situ* data such as still images, videos, or samples of the area in question. It is the relationship which is inferred between the different data sets inferred using machine learning techniques that varies between studies. A considerable portion of such studies are shown to use deterministic methods to predict a label for any given coordinate such as Random Forests and Support Vector machines (SVMs), whilst more recent ones make use of more informative methods such as Gaussian Processes, providing a distribution over all possible labels given any data point.

1.1 Contribution

The main contribution of this thesis will be to explore how to use data where a single data point does not only have one label exclusively, but instead corresponds to a tally of each possible label. For example, a particular 5m x 5m area in the benthos may be an even mix of both sand and coral, but in previous literature, the data was simplified such that whichever

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

label occurred more frequently regardless of how small the margin would be the single label assigned to that point. This results in a very coarse approximation even when using Gaussian Processes attempts to model the uncertainty/uncertainty with its predictions at each point (but ultimately only provides a single, final prediction). To alleviate this and provide a richer set of information, we explore the use of Dirichlet Multinomials, which provides a distribution of each label that represents something entirely different. Whereas in a Gaussian Process, each label is assigned the probability of being the correct one, the output of a Dirichlet Multinomial Regressor provides the distribution of the frequency of labels in a particular space itself. See section GP vs DM for an illustrative example on how results would differ in practice between the two methods.

1.2 Motivation

The motivation behind assessing the effectiveness and advantages of such a method are that they inherently tie in with lower resolution data, particularly when a single images corresponds to a large enough area such that one would expect a mix of different labels. This is advantageous because we want to be able to re-sample data from any given site periodically (for example, every 3-4 years) whilst being economically efficient. This naturally lends to lower resolution data, meaning that summarising large areas to a single label would theoretically be throwing away a majority of the information contained in bathymetry and image data.

1.3 Outline

We will first look at the existing literature in chapter 2 - on 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 contrast 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 chapter 2.

CHAPTER 2

Literature Review

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 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, we will give a brief overview of each of these steps, along with common procedures involved.

- **Habitat Characterisation** - extracting properties of the environment such as rugosity (roughness), aspect (direction of slope), depth
- **Habitat Classification** - grouping the raw information about the environment into categories, such as sand, granite, etc.
- **Habitat Mapping** - using classifications with the larger scale bathymetry data to extrapolate habitat maps

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.

2.1.1.0.1 Remote-sensing data. 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, which involes the firing of sound waves towards the benthos, where their frequency and strength upon returning is used to deduce the physical properties of the area from

[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.

where it rebounded, such as depth, slope, and roughness. More modern acoustic backscatter collection methods like Multibeam echosounders (MBES) are gradually superceding older ones such as Single-beam echosounders (SBES)(Calvert et al., 2015) due to the *cone* of the beam angle of the sound pulse in MBES being of a considerably smaller angle to reduce the amount of coarse, repeated data needed to try and resolve the true properties contained. (Brown et al., 2011). At the same time, it maintains a larger swath angle, allowing data to be collected at a fast rate.

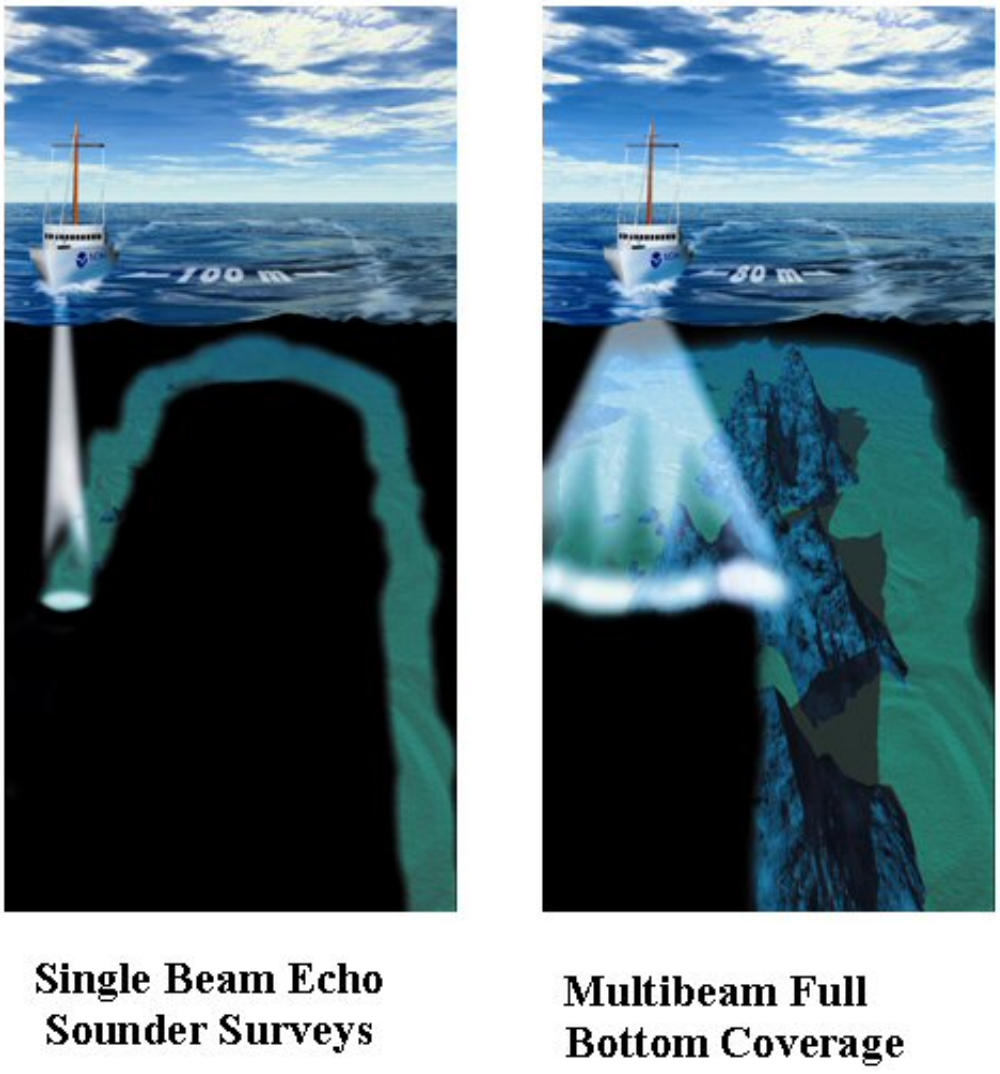


FIGURE 2.1: Rough comparison of SBES with MBES (Marine, 2016)

As pointed out in (Calvert et al., 2015), bathymetry data alone is not enough and should not be the sole basis on which habitat maps are formed, and as such, more conclusive information is required to make predictions in these studies.

2.1.1.0.2 Truthing Data. One of the more common methods to be able to obtain a sufficiently large truthing data set (but small relative to areas with remote-sensing data) are videos or images, generally by autonomous means, where images have the advantage of not requiring the additional post-processing that video does to extract the images. Technology to send unmanned vehicles to the benthos and capture visual data is a relatively recent development, and prior to that, it was more common to send divers on manual expeditions that involved actually collecting sediment samples that were used to identify what the habitat at a particular location may have been.

2.1.2 Habitat Classification

more in-depth focus here, what kind of supervised/unsupervised ML algorithms are used for classification? Almost all studies use *in situ*[2] 'truthing' data to complement the acoustic data to be able to build a model between the acoustic data and truthing data (creation of these models are explained in following sections). However, we need to know the labels of this data considering that the final goal is to create a habitat map, where any one habitual zone is given its prospetive label - to do this, we also need to label the clusters of truthing data. These categories may be, for example, 'bedrock covered by discontinuous seagrass cover', 'Maerl interspersed with sand and gravel', 'superficially coarse sand to fine gravel covered by dense patches of seagrass', etc. (Micallef et al., 2012). Given physical sediment truthing data, the process of labeling locations is inevitably a manual one, but this is not the case when dealing with image data. The two overarching ways to perform this classification with images are using supervised and unsupervised algorithms.

[2]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.

Studies have used both unsupervised clustering and supervised clustering (classification) 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 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[3] 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.

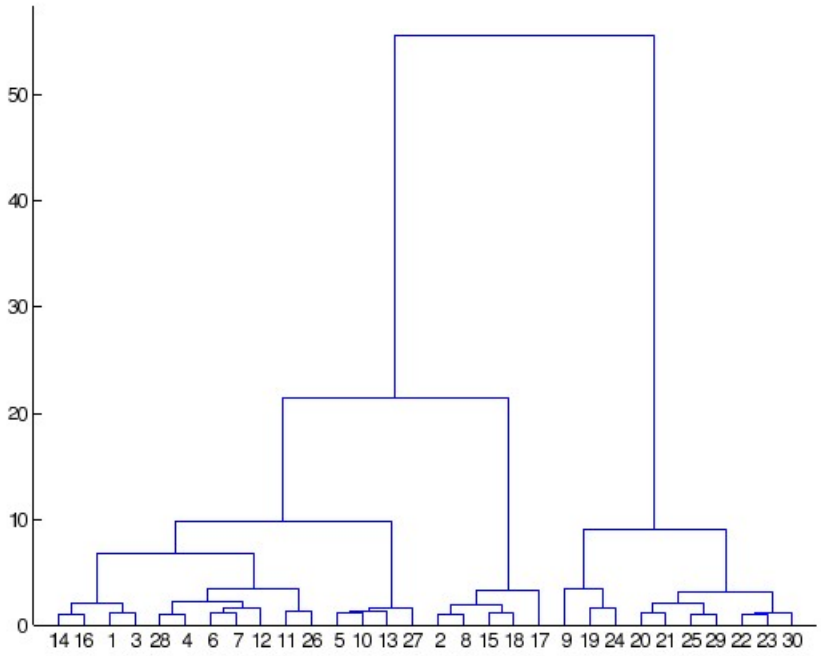


FIGURE 2.2: Dendrogram of images from Pizarro et al. (2009) using Kullback-Leibler convergence distances between feature distributions. Clusters in the lower layers were ommitted to prevent visual clutter.

[3]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.

2.2 Non-Machine Learning Methods

this section should be part of the previous one (Map Creation) 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[4] 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 employ machine learning techniques to further utilise the information that we gather.

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

2.3 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 mathematical background will be given where relevant in the following algorithms, as their application will be revisited in chapter 5.

2.3.1 Multiple Multinomial Logistic Regression

The underlying function of logistic regression is that used in linear regression, so a brief overview of linear regression will first be given. The motivation for linear regression is that given some input data $\mathbf{X}$, corresponding response values (or output) $\mathbf{y}$, and a set of weights β are defined to best define the relationship between the two variables as below. Note that the intercept term has not been included in this basic explanation, as it can be included by augmenting the inputs $\mathbf{X}$ with a bias term 1 which would have its own corresponding weight β_i as well.

$$\begin{aligned} \mathbf{y} &= \beta\mathbf{X} \\ y_i &= \beta_i x_i, i = 1, ..., n \end{aligned} \tag{2.1}$$

To find these set of weights, we need to optimise this equation with respect to weights β such that $\beta\mathbf{x}$ is as close to $\mathbf{y}$ as possible. In other words, we wish to minimise the errors, or residuals. Once they have been found, the weights are then applied to any future inputs with unknown response values (extra details on optimisation here). The restriction of this only finding linear relationships can be lifted by projecting inputs into higher dimensions, which allows 'linear' regression to then model more complex data.

To then perform multiple multinomial logistic regression, we need to first manipulate the multi-class label data into binary labels, such that for some set of n discrete labels $1, 2, ...$, we consider n cases where for the $i-$th case, i takes on the value 1, whilst all others becomes 0 (see Example 2.3.1 for an example). This is needed as logistic regression only allows predictions of binary labels - this corresponds to a one-vs-all logistic regressor. However, this doesn't guarantee that predictions that each underlying linear regressor will fall within the $[0, 1]$ boundary. To ensure this, we use the logistic function (Figure 2.3) to 'clamp' all response variables back between 0 and 1, where α is a single prediction, or a vector of them:

EXAMPLE 2.3.1.

$$1, 1, 1, 2, 2, 2, 3, 3, 3$$

$$\text{1-vs-all: } 1, 1, 1, 0, 0, 0, 0, 0, 0$$

$$\text{2-vs-all: } 0, 0, 0, 1, 1, 1, 0, 0, 0$$

$$\text{3-vs-all: } 0, 0, 0, 0, 0, 0, 1, 1, 1$$

$$\text{logistic}(p) = \frac{1}{1 + exp(-\alpha)} \tag{2.2}$$

This will then provide a vector of predictions with the same length as the test inputs for *each* label, representing a probability that that particular data point is of that label. To simplify this

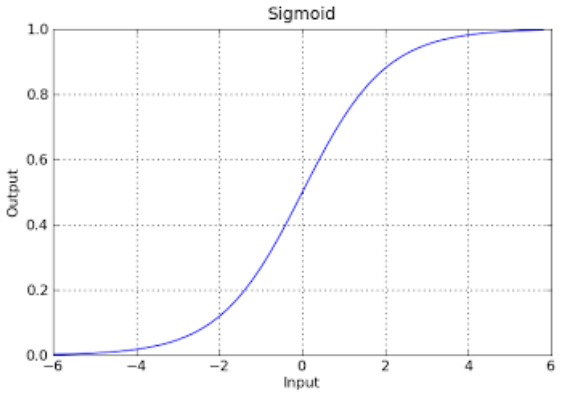


FIGURE 2.3: Logistic Function

into a set of output labels as desired, one would simply need to take the argmax of predictions per label. For example, given these raw predictions for a dataset with 4 possible output labels, $1, 2, 3, 4$:

$$\begin{bmatrix} 0.005 & 0.370 & 0.314 & 0.150 & 0.559 & 0.512 \\ 0.492 & 0.236 & 0.911 & 0.540 & 0.539 & 0.383 \\ 0.536 & 0.435 & 0.647 & 0.154 & 0.511 & 0.737 \\ 0.684 & 0.773 & 0.670 & 0.667 & 0.990 & 0.944 \end{bmatrix}$$

The max values per column are: $0.5590.9110.7370.990$, corresponding to the labels $3, 3, 1, 3, 3, 3$.

Multiple Logistical Regression is one of the more basic machine learning algorithms that can be used to predict habitat classes. Regression, broadly, involves the estimation of relationships between variables, and logistic regression involves the prediction of likelihood of class membership given a number of variables (that are assumed to have low collinearity[5]). This only applies to domains with two classes, however - to use this technique for classification where we have an unbounded number of classes (though usually still relatively low), we need to use multinomial logistic regression, which is able to account for more than two distinct, unordered (i.e., sand vs. mud has no relative ordering) classes, where class membership is predicted using

[5]Collinearity between two variables refers to how correlated they are - a contrived example would be measurements of depth taken in centimetres and metres.

maximum likelihood estimation (MLE), similarly to logistic regression. However, the difference is that whereas logistic regression only requiring a single logit function as the variable being predicted is a binary one, multinomial logistic regression requires comparison between $k-1$ (where k is the number of possible dependent variables) logit functions.

Even though Caruana and Niculescu-Mizil (2006) show that logistic regression methods achieve on average worse results than most other approaches available, it recognises that in certain cases the models that perform most poorly on average still display exceptional performance, and as such, this method is still worth exploration and experimentation. In particular, Belanger et al. (2012) used multinomial logistic regression across temperature, salinity, and productivity to correctly predict class membership by a margin of 23-84% more than by pure chance. This is equivalent to an improvement of 1-2x compared to a random guess, which taken at face value would suggest that logistic regression is an undesirable choice of algorithm for this problem domain.

2.3.2 Random Forests

In contrast to logistic regression, random forests were shown in Caruana and Niculescu-Mizil (2006) to be state of the art, only just falling short of boosted decision trees after callibration. Random forests are an ensemble method, meaning that it uses a collection of estimators, before aggregating their results to obtain some sort of average. The aim of this is to minimise the variance and hence error that any single one of these estimators would otherwise result in.

From the initial dataset, some number B is chosen which represents the *number* of trees to build (as a part of our random forest), after which, B random, unique subsamples of the full dataset are taken. Within each decision tree in our random forest, some constant number m of features is taken at each node of the tree, such that the split at each node only takes into account the m randomly chosen features. Each of the decision trees in our forest will hence have a 'result' (that may be a class or some continuous value). Typically, the final decision of the random forest will be made by a vote count for classification, and an average of each decision tree's result in regression problems.

As random forests are a method that is low in complexity but provides very good results on average, we can see that it is used in quite a few studies (Lucieera et al. (2013), Seiler et al. (2012), Hasan et al. (2014)), where the random forest classifier provided the best results over other methods relating to at least a significant subset of the explored data. 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. 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.

2.3.3 Multi-class Support Vector Machines

Multi-class support vector machines require similar additional steps to those required by logistic regression to be able to do multiple (multi-class) multiple logistic regression. SVMs also only inherently support two classes when used for classification.

In Figure 2.4, we see an SVM find a plane (or line, in the 2D case) that maximises the distance to the closest point of each of the other two classes. To generalise this to any number of classes greater than 2, one option is to consider a one-vs-one approach in contrast to the one-vs-all one in Example 2.3.1. In this case, an SVM is build for all pairs of labels, meaning that for c unique labels, $\frac{c(c-1)}{2}$ separate SVMs are needed. When performing predictions, a tally is kept for each input data row counting how many of each class they were predicted to be by all the SVMs, with the highest count being the final prediction.

Although support vector machines fell outside the top three overall supervised learning algorithms in terms of accuracy in Caruana and Niculescu-Mizil (2006)'s work, their non-parametricity could potentially be of benefit given that our knowledge of the complex relationships between

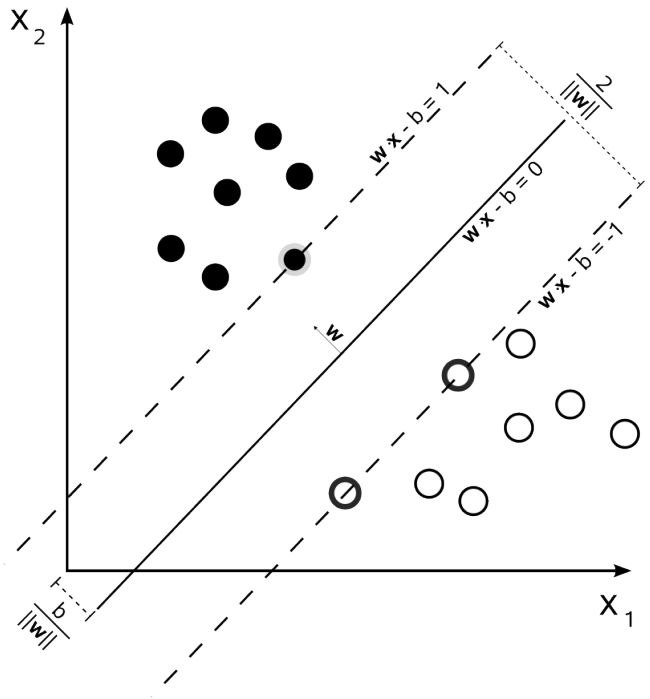


FIGURE 2.4: Basic example of an SVM

elements of benthic habitats are limited. Moreover, despite SVMs not being used often in the field, they are "acknowledged to be very competitive descriminative classifiers in machine learning literature" (Ahsan et al., 2011).

Multi-class SVMs were used in Ahsan et al. (2011), using the one vs. one approach as per their use of LibSVM (Chang et al., 2011), outperforming classification trees on certain datasets, and in limited cases (but not overall), Gaussian Mixture Models as well. Again, the mixed results would suggest that under certain conditions (such as size of the dataset and various properties of the data itself, some of which are not known before testing), use of a multi-class SVM could provide a useful benchmark to some extent.

2.4 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 isn't 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% coral, and 74% to be 25% coral' (with the remaining probable distributions excluded for simplicity). These techniques are the focus of the next chapter.

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 exception to this was logistic regression, but even then, as a parametric method, the complexity of the model 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 technique to generate predictive habitat maps. We begin by visting Gaussian Process Regression, and how a small extension/post-processing step extends it to allow Gaussian Process Classification. Of the different ways to train the hyperparameters of Gaussian Processes, the one selected was Leave-One-Out Cross Validation (LOO-CV). They also need to use a *kernel* that defines the relationship between any two points, forming the full covariance matrix - the one chosen was the squared exponential kernel, explained in detail in the following sections. Note that detailed proofs and derivations are not covered here, and interested readers should consult Rasmussen and William's Gaussian Processes for machine Learning (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 standard linear regression 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 mean (m) and covariance (k), 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 (on the right hand side). The base covariance function between points p, q, where $\mathbf{x_p}, \mathbf{x_q}$ are the vector of n features at each point is thus given by:

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

From the above equation, it is evident that points that are very close together in the n-dimensional input space would have a covariance of 1 (as $\lim_{\mathbf{x_p}=\mathbf{x_q}} exp(-\frac{1}{2}|\mathbf{x_p} - \mathbf{x_q}^2| = 1)$ - when assuming all features are equally important and correlated when assessing their distance. Because logic would define that such an assumption is unlikely to hold with real world data, a *length-scale* needs to be applied to each dimension to give important features more weight, and reduce the impact of less significant features on the covariance between two points. The vector of lengthscales would then be optimised along with the other parameters when training the Gaussian Process model. Should 3.2 prove to be the optimal setup, the length scale vector would then simply comprimse of 1s after training. The updated covariance function k would then be:

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

where σ_f is the variance in the training data, and σ_n is the variance of the Gaussian noise. clarify what's happening here

To allow simplifications of notation in the following equations, we define some abbreviations related to Equation 3.3 depending on what data is involved in the covariance matrix.

(lay out these abbreviations nicely) To indicate the full covariance matrix over training points: $K = K(X, X)$
To indicate the full covariance between training points and test points: $K_* = K(X, X_*)$ To indicate the covariance between a single test point with all training points: $\mathbf{k}_* = \mathbf{k}(\mathbf{x}_*, X)$

By conditioning the joint Gaussian prior distribution on the observed data, we obtain our predictions at test points (EXPAND. Possibly include non-abberviated version first for clarity):

$$\mathbf{f}_*|\mathbf{x}_*, X, \mathbf{f} \sim \mathcal{N}(\mathbf{k}_* K^{-1}\mathbf{f}, k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_* K^{-1} K_*) \tag{3.4}$$

Taking the first part of the Gaussian Distribution, the mean, and the second, the variance, we obtain our predictions and variance on a single test point:

$$\bar{f}_* = \mathbf{k}_*{}^T (K + \sigma_n^2)^{-1}\mathbf{y} \tag{3.5}$$

$$\mathbb{V}[f_*] = K(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*{}^T (K + \sigma_n^2)^{-1}\mathbf{k}_* \tag{3.6}$$

In practice, a test dataset would not be calculated one test point at a time as the above equation suggests, but all at once - to do so simply requires taking the covariance between all test points and training points whenever covariance of only a single test point with all training points is involved. These equations are used on the basis that all their parameters have already been determined - the most common process for doing this and the one used in this study is to maximise the log marginal likelihood. Although optimising the log marginal likelihood is required to use the above equations, its representation requires notation defined above, and as such formalising this aspect has been withheld until this point. However, because a slightly customised log likelihood function is used in Leave-ont-out Cross Validation (LOO-CV) below, we will only cover the standard marginal likelihood used by Gaussian Processes briefly here.

The marginal likelihood for this model can be obtained by integrating the likelihood times the prior with respect to function values $\mathbf{f}$:

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f}, X)p(\mathbf{y}|X)d\mathbf{f} \tag{3.7}$$

Taking the log of this then gives:

$$\log p(\mathbf{f}|X) = -\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{3.8}$$

As this equation has an analytical first derivative which can be used to speed up optimisation so long as a software package is used where the optimisation algorithms allows a derivative (also known as a Jacobian), we wish to also know its derivative:

$$\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|X,\theta) = \frac{1}{2}\text{tr}((\alpha\alpha^T - K^{-1})\frac{\partial K}{\partial\theta_j}) \text{ where } \alpha = K^{-1}\mathbf{y} \tag{3.9}$$

Keeping in mind that K initially contains a number of initially unknown parameters (σ_n, σ_f, l), we can then optimise over this log marginal likelihood function as a whole using its derivative to search the multi-dimensional space. However, a variant of this will be used, as explained in the following section.

3.2 Leave-One-Out Cross Validation

(expand section)

To train our data, we chose the extreme case of of cross-validation for model training, where the number of folds used, k, is equal to the number of datapoints. 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. The latter case is preferrable here as biological experts

were not consulted for the duration of the study, meaning some assumptions could have been tuned more accurately if external help was available.

The log probability ommitting training case i

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

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

LOO-CV predictive mean and variance

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

partial derivatives with respect to the hyperparameters

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

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

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

3.3 Gaussian Process Classification

To perform Gaussian Process Classification, multiple Gaussian Process Regressors are used for each possible label using a one-vs-all approach as in Example 2.3.1, and every label at each data point is assigned both a probability and variance. To ensure the predictions at each point for each label is propertly constrained to be in the range $[0, 1]$, we again pass predictions through the logistic sigmoid function (Equation 2.2).

(expand)

3.4 Gaussian Process Approximation

However, significant limitations exist in terms of the number of training datapoints used when using a single Gaussian Process Classifier. To counter this, there are approximation methods which allow more points to be used whilst not expending more time. There are two common ways to do this, the first being using inducing inputs, where approximations are made such that neither extra time or computational power is needed to encapsulate the information of more points. The second are *ensemble* methods - combining the results of several independent Gaussian Processes trained in parallel (at the expense of more computational power), and was the one tested as a part of this study.

3.4.1 Product of Experts and Variations

(EXPAND explanations in this section)

Product of GP Experts

$$\mu_*^{poe} = (\sigma_*^{poe})^2 \sum_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{3.10}$$

$$(\sigma_*^{poe})^{-2} = \sum_k \sigma_k^{-2}(\mathbf{x}_*) \tag{3.11}$$

Generalised Product of GP Experts

$$\mu_*^{gpoe} = (\sigma_*^{gpoe})^2 \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{3.12}$$

$$(\sigma_*^{gpoe})^{-2} = \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*) \tag{3.13}$$

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 maintain reasonable margins of error.

3.4.2 Bayesian Committee Machines and Variations

Bayesian Committee Machine

$$\mu_*^{bcm} = (\sigma_*^{bcm})^2 \sum_{k=1}^{M} \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{3.14}$$

$$(\sigma_*^{bcm})^{-2} = \sum_{k=1}^{M} \sigma_k^{-2}(\mathbf{x}_*) + (1-M)\sigma_{**}^{-2} \tag{3.15}$$

where σ_{**}^{-2} is the prior precision of $p(f_*)$, which itself is the inverse of the prior variances.

Robust Bayesian Committee Machine

$$\mu_*^{rbcm} = (\sigma_*^{rbcm})^2 \sum_{k} \beta_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{3.16}$$

$$(\sigma_*^{rbcm})^{-2} = \sum_{k} \beta_k \sigma_k^{-2}(\mathbf{x}_*) + (1 - \sum_{k=1}^{M} \beta_k)\sigma_{**}^{-2} \tag{3.17}$$

where each β_k follows the same rules as for the Product of Experts and its variations.

(Deisenroth, 2015)

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 the Dirichlet Multinomial Distributions that have the ability to perform predictions over category counts, a perfect fit for the original data used 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 first provide a brief overview of the multinomial and dirichlet distributions separately.

explain why we should first revisit dirichlet, multinomial distributions separately before looking at dirichlet multinomial regression

4.1 Multinomial Distribution

equations, description

4.2 Dirichlet Distribution

descriptions

$$\theta \sim Dir(\alpha) \text{ , dirichlet distributed random variable}$$

$$p(\theta) = \frac{1}{\beta(\alpha)} \Pi_{i=1}^{n} \theta_i^{\alpha_i - 1} I(\theta \in S) \text{ density function, I is indicator function}$$

$\theta = (\theta_1, ..., \theta_n), \alpha = (\alpha_1, ..., \alpha_n), \alpha_i > 0$ theta - n-dimensional vectors, alpha - parameters for distribution

$S = \{x \in R^n : x_i \geq 0, \sum x_i = 1\}$ S is probability simplex, the set of pmfs on numbers 1 through n

$$\frac{1}{\beta(\alpha)} = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_i)...\Gamma(\alpha_n)}, \alpha_0 = \sum_{i=1}^{n} \alpha_i \text{ generalised beta function}$$

4.3 Dirichlet Multinomial Regression

descriptions

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

$$M = \sum_k c_k$$

For the regressor, the two activation functions that were considered were exponential and softmax, where the former often provided better mapping predictions, but the latter is preferable in the general case due to its better numerical stability include graphs of exponential and softmax here.

$$\alpha_k = \exp\{x^T w_k\}$$

$$\alpha_k = \text{softmax}\{x^T w_k\}$$

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 likelihood then then gives the posterior over which to optimise to obtain the weights required to predict the normalised label counts at any given point.

This gives the joint-log-likelihood over both the dirichlet and multinomial distributions:

$$\sum_{n=1}^{N}[\log(M_k) - \sum_k \log(c_k!) + \log\Gamma(\sum_k \alpha_k(x_n)) - \log\Gamma(\sum_k c_{nk} + \alpha_k(x_n))] + \sum_{n=1}^{N}\sum_{k=1}^{K}[\log\Gamma(c_k + \alpha(x)) - \log\Gamma(\alpha_k(x_n))] + \sum_{k=1}^{K}[-\frac{\phi}{2}\log(2\pi\phi) - \frac{1}{2}w_k^T\phi\mathbb{I}w_k] \quad (4.1)$$

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

$$\partial\frac{\log p(c,x)}{\partial w_k} = \sum_{n=1}^{N} x_n\alpha_k(x_n)[\psi(\sum_l \alpha_l(x_n)) - \psi(\sum_k c_{nk} + \alpha_k(x_n))] + \sum_{n=1}^{N} x_n\alpha_k(x_n)[\psi(c_{nk} + \alpha_k(x_n)) - \psi(\alpha_k(x_n))] - \frac{1}{\phi}w_k \quad (4.2)$$

explain all the symbols here

4.3.1 Using Markov Chain Monte Carlo (MCMC) instead of MAP

(give extra info on MCMC here)

Using the MAP for optimisation, while a valid option, ignores all the possible weights in the given posterior distribution where they could also be drawn from. To create more certainty that the weights drawn from the distribution are in agreeance about the actual maps generated, MCMC would need to be used. By applying weights from the MCMC chains back to predictions and visually comparing differences, visual confirmation can be had on whether the maps remain reasonably consistent using weights drawn from the posterior.

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. Note that three clusters were synthesised, with clusters A, B containing $0.7 : 0.3$ and $0.3 : 0.7$ average ratios in label mix per point respectively, while cluster C contained an even $0.5 : 0.5$ average split, where cluster had 100 points. The colours on the overall plot are only representative of the **most** common label at each point - the actual distributions at each point are shown in the graphs following it.

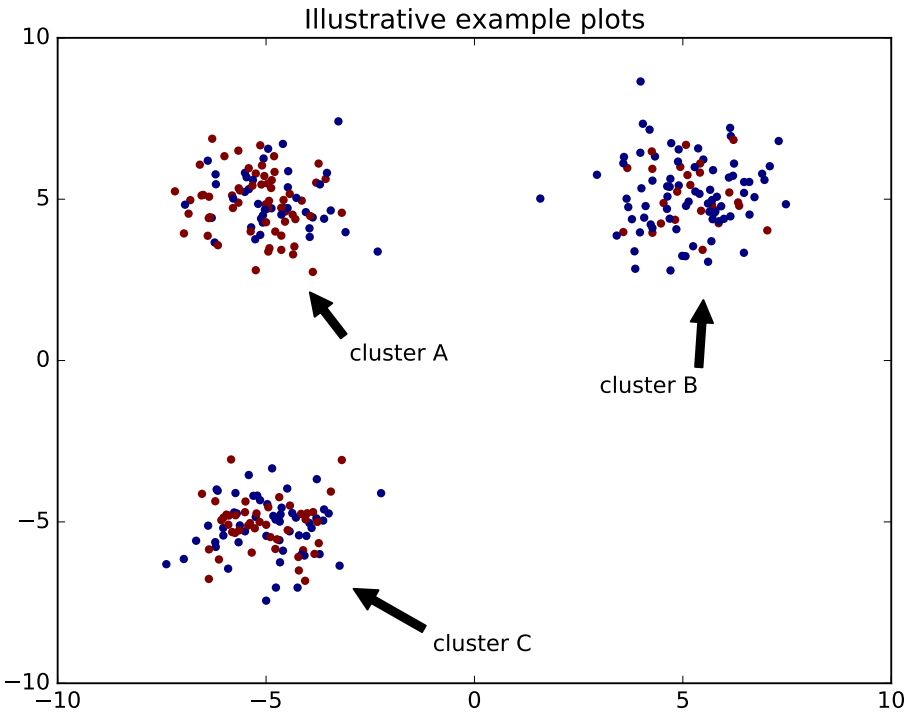


FIGURE 4.1: Plots of the three clusters, with labels taking on the argmax of each point

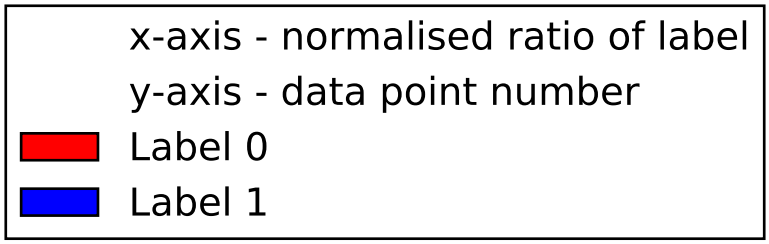


FIGURE 4.2: Legend/axes for the following histogram plots showing distribution of labels at each point

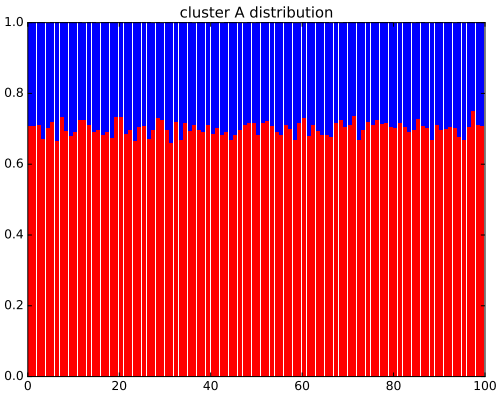


FIGURE 4.3: Label distribution of cluster A

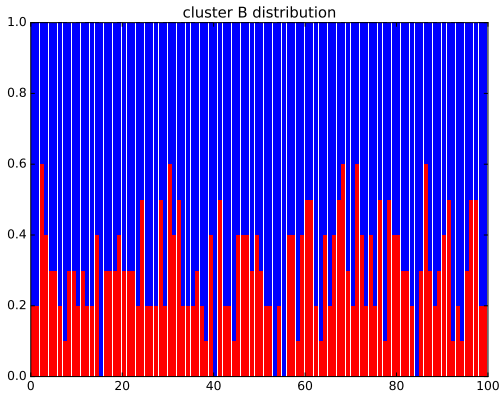


FIGURE 4.4: Label distribution of cluster B

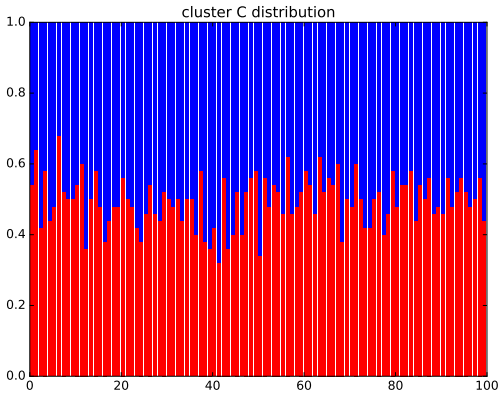


FIGURE 4.5: Label distribution of cluster C

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. In an initial attempt to counter this, multi-task GPs were considered as a means of making a *fairer* comparison between a GP and DM, but the idea was ultimately discarded as it was not fit for purpose, one of the primary issues being that the model does not inherently restrict the outputs of a given datapoint to sum to 1, instead being at the mercy of the parameters of the GP.

4.4.1 Results

The results and plots for this example are below, and figures displayed were taken from an average of 20 runs.

	Dirichlet Multinomial Regression RMSE*	Gaussian Process Classifier (argmax) RMSE
Original data	0.070179271314358999	0.26833333333333337
Quadratic-space projection	0.065630111843395234	0.43433333333333335
Cubic-space projection	0.29019235800882354	0.43725490196076466

RMSE - root mean squared error

As can be seen from the above overvise, the DM performed best when projecting the data to quadratic space, while the GPC didbest 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. Note that the exact probabilities provided by the GP are hown in the following plots, in contrast to the argmax taken for error-calculation purposes. (don't use these graph plots here, do per-label heatmaps)

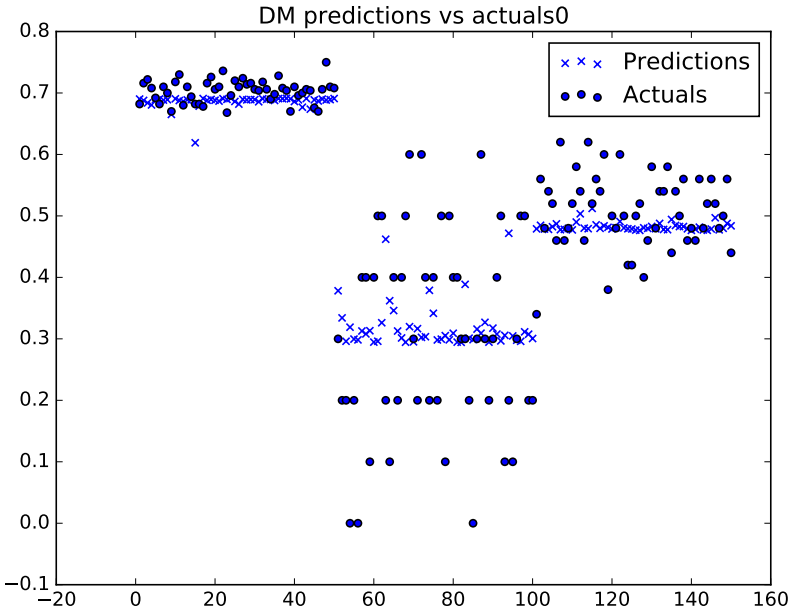


FIGURE 4.6: DM Label distribution of label 0

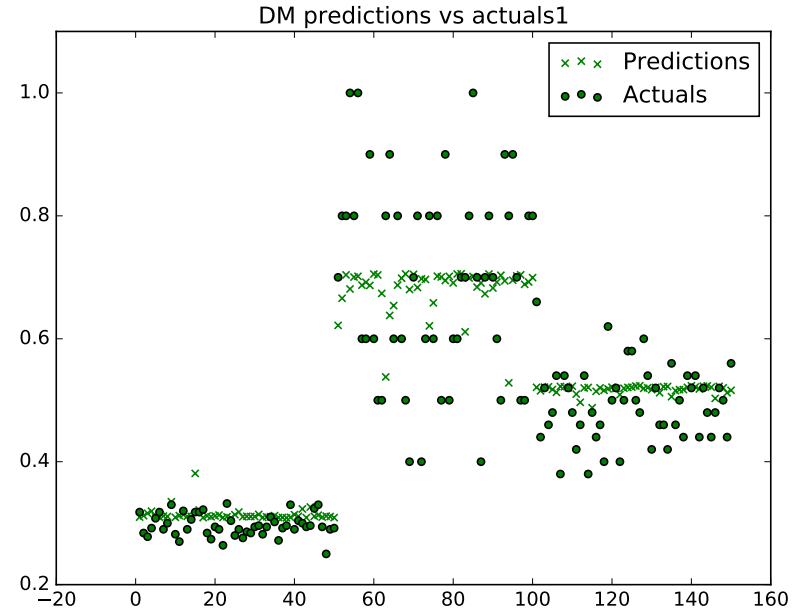


FIGURE 4.7: DM Label distribution of label 1

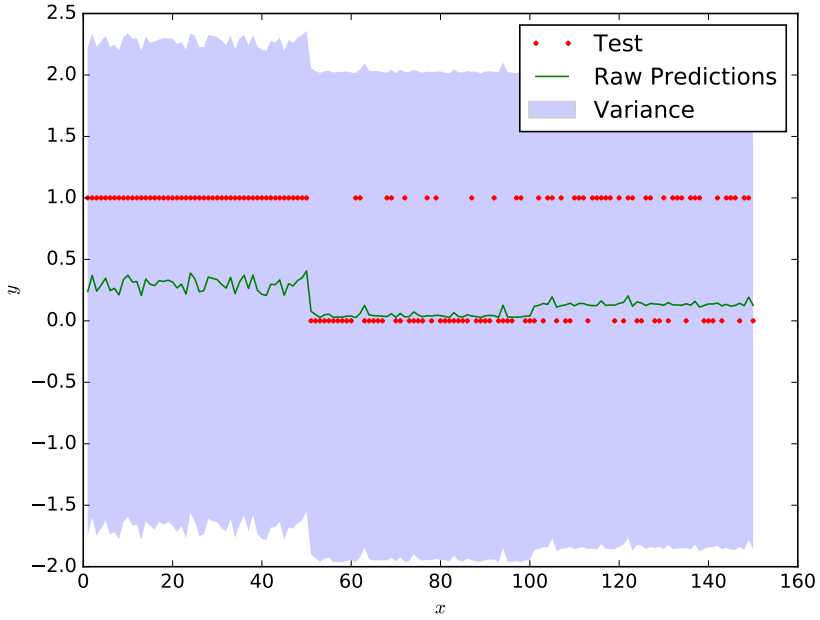


FIGURE 4.8: OvR GP performance with variance for label 0

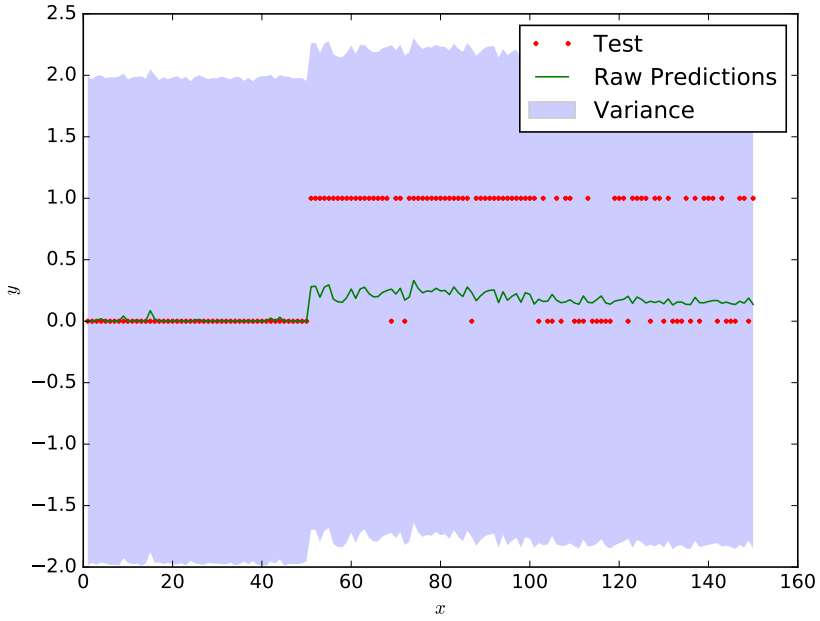


FIGURE 4.9: OvR GP performance with variance for label 1

As we can see, the DM performed notably better than the GP, with the predictions in each cluster staying true to the mean values. The GP, despite the ~ 0.26 error rate, can be seen to follow a consistent trend in the test data instead of adjusting to the two different classes. Most importantly, for both label 0, 1, the DM identifies that the portion of either label in cluster C is 0.5, due to its ability to learn and predict distributions at a point. As GPs are not designed to and cannot model this, it instead finds that neither label has a high probability, whilst also having a very high variance (beyond the $[0, 1]$ bound which again, the DM enforces but is not a property of a GP).

However, this is admittedly admittedly a rather simple example that assumes we have a sufficient amount of data from the *three* possible habitat clusters - A by itself, B by itself, and a homogeneous mix of A and B, and as such, more detailed comparisons will be made using the full training dataset.

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 more confident in the fact that that area does in fact have a mix of labels.

[H]

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 (the reef in the centre of Figure 5.1). The bathymetry data was collected using Eric Schmidt's Falkor at the locations in Figure 5.2, a ship dedicated to marine research, and the (over 700GB) of images corresponding to all the bathymetry data was collected by The University of Sydney'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). (currently not relevant/no context for next sentence) On close inspection, the UTM 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 weren't 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. (describe the specific features - depth, aspect, rugosity, etc.)

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 (calculate how much perhaps). To do this, a 'grid' of squares was essentially mapped onto 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 was selected randomly *but* with more weighting to the non-sand labels, to slightly increase the variance of labels observed. The reason for doing this was because of the dominance of the 'sand' habitat - random selection when only one absolute label per point was present resulted in an even larger portion of sand than in the original dataset, causing predictions on the full query data to sometimes result in almost 100% sand, even moreso than was already the case in Figure 5.10.

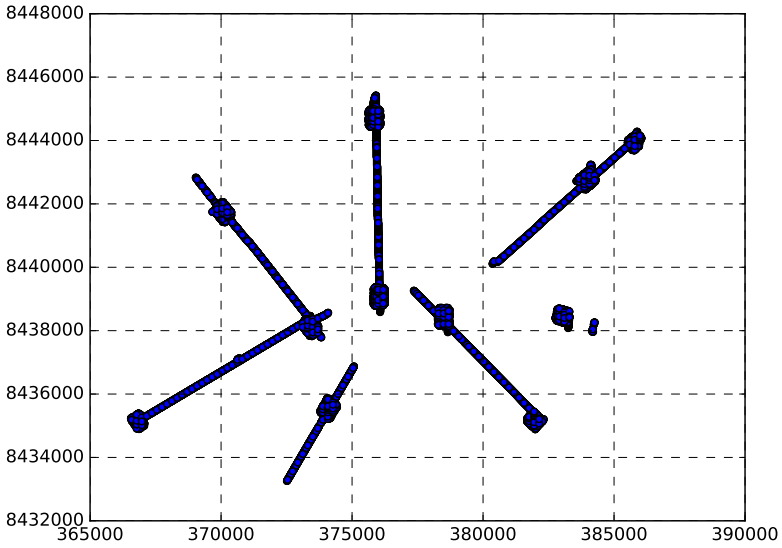


FIGURE 5.2: Fixed-sized grids placed over training data (redo this plot nicely!)

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 determined through an automated clustering procedure using Dirichlet Processes. Because of the uneven distribution of these labels (Figure 5.6 and Figure 5.7), 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. Moreover, due to the unsupervised nature of the labeling, certain clusters were notably *inconsistent* with the rest, for example when sea cucumbers became the identifying feature of one of the 24 labels.

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

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

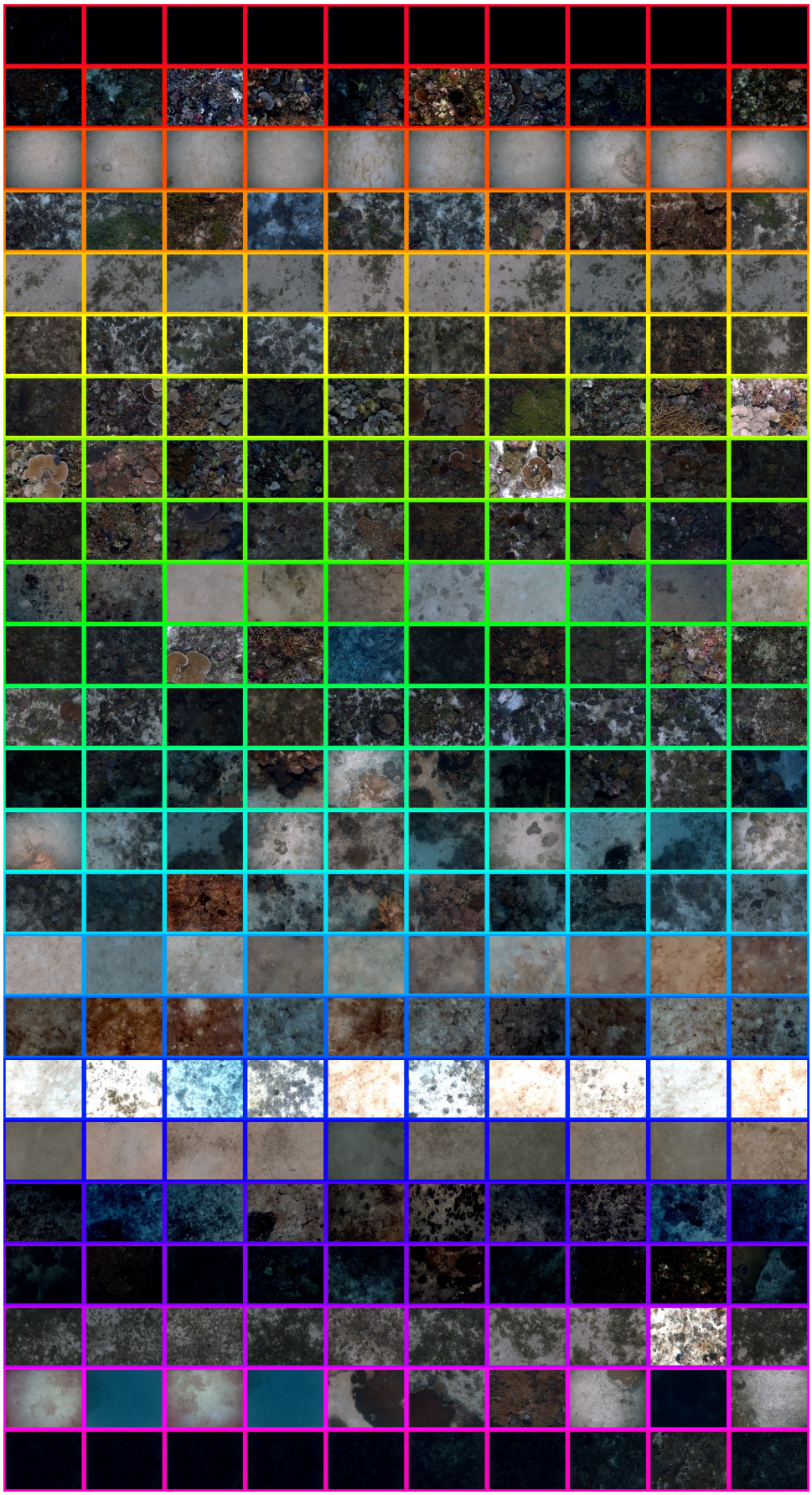


FIGURE 5.3: Samples of images from each of the full 24 classes mark with the simplified labels adjacent to it

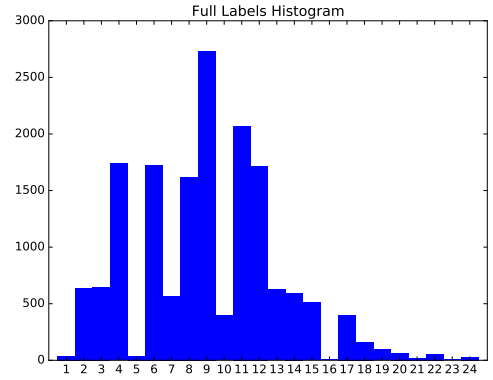


FIGURE 5.4: Distribution of labels in original dataset

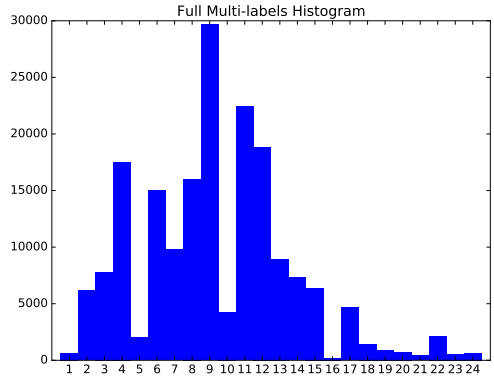


FIGURE 5.5: Distribution of labels in multi-label outputs

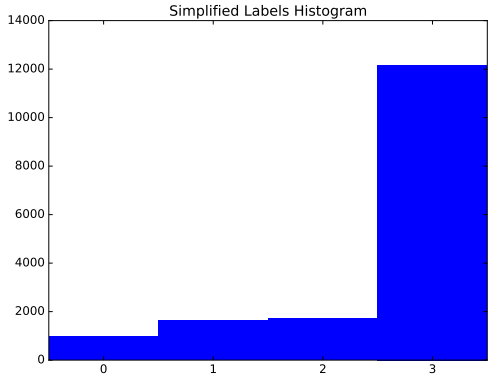


FIGURE 5.6: Distribution of simplified labels in original dataset

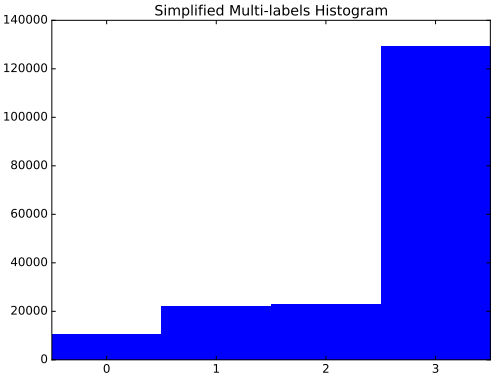


FIGURE 5.7: Distribution of simplified labels in multi-label outputs

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 Coordinates as features

Due to the abundant bathymetry data that was available in the form of depth, rugosity and aspect at each available data point, the coordinates themselves were not included in the feature space for experiments. Whilst it is logical that in a natural environment, areas that were spatially close would also have similar properties, this should not be relied upon, and other intrinsic properties should be the basis upon which predictions are made. Forming predictions on the full query dataset using a random forest supports this notion quite strongly - whilst 10-fold cross validation using the coordinates as features had a notably higher F-score of 0.61 compared to 0.40 without, the unnaturally straight split between the left and right segments (Figure 5.8) over a 12km region suggests that the predictive map is flawed. Moreover, by including the coordinates as a training feature, an assertion is made about the direct relationship between a benthic location and the habitat class/es it contains, despite its other physical properties such as depth, aspect, etc - an assumption that should not be embedded into the model before fitting it to the data even begins.

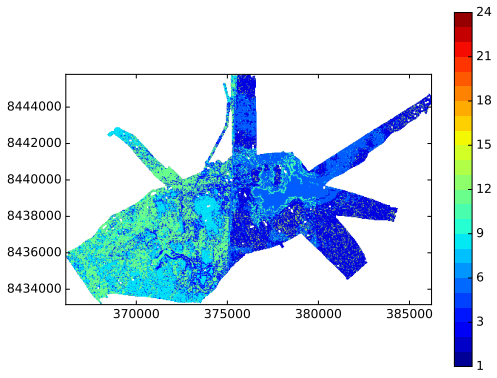


FIGURE 5.8: Full predictive map using Random Forests including coordinates as features

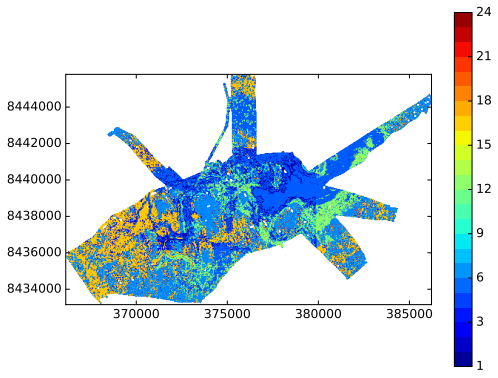


FIGURE 5.9: Full predictive map using Random Forests excluding coordinates as features

5.2.4 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 (include plots of the diff approaches across DM/GP/others, ref plots). 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, one of the reasons being that it resulted in an lower average error when performing 10-fold cross-validation using Dirichlet Multinomial Regression (Table 5.5). 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.

(ref plots)

(show plots)

5.3 Single-Output Predictions

5.3.1 Deterministic Approaches

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

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

$2 * \frac{\text{precision*recall}}{precision+recall}$. 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 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. Logistic regression has been included here despite containing 'probabilistic' predictions in the form of regression values passed through the *logit*, and as such the results displayed are a result of simply taking the argmax over possible the predictive probability over possible for each datapoint. Those probabilistic outputs are useful for comparisons with those of Gaussian Processes, however, which will be explored in the next section. (this GP-LR comparison hasn't happened yet)

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 5.10 visualises the predictions from Table 5.2 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.

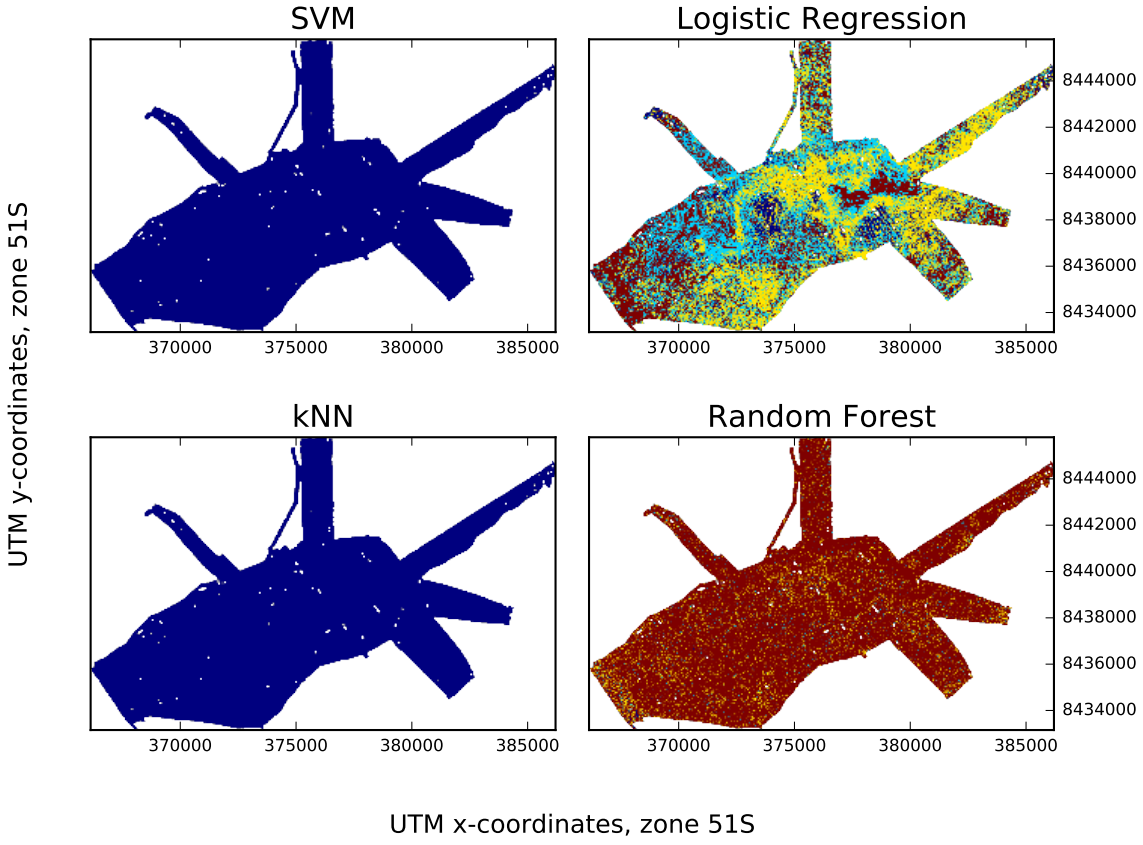


FIGURE 5.10: Full predictive map using SVMs, Logistic Regression, kNN, and Random Forests

The maps in Figure 5.10 (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.

5.3.2 Probabilistic Approaches

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.

5.3.2.1 Single-Output 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 *likelihood* 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.

Area Under The Receiver Operating Curve TODO

show more stratified results (not just even split) to show that even splits did better

Type of GP Classification	AUROC	Accuracy	F-score	Labels used
Normal	TODO	0.75525	0.47921	4 labels
PoEGP				
GPoEGP				
BCM				

No. points	Type of split	Type of GP	Number of runs	AUROC	Notes	F1-score
500	Even	GP	10	0.86534		
500	Stratified	GP	10	0.80136		
1000	Even	GP	1	0.87626	Deterministic	0.56208
1000	Even	PoEGP	5	0.80973		0.47481
1000	Even	PoEGP	200	0.80186		0.47595
1000	Even	GPoEGP	5	0.80864		0.51018
1000	Even	GPoEGP	200	0.80105		0.47748
1000	Even	BCM	5	0.80682		0.48167
1000	Even	BCM	200	0.80421		0.48227
1000	Even	GPy	1	0.87638	RBF, EP (default)	0.57013

(look at AUROC/AUC and log probabilities as well)

highlight areas with low/high certainty, etc. NOTE - investigate the areas with visually even splits of two labels - e.g. right-side arms of label 1,2, and smaller patches in the bottom left corner of label 0,3 - show that uncertainty about whether those areas are label 1 or 2, 0 or 3 respectively, is (should) be high, and that taking argmax for the sake of visual representation within a single image hides this information

(talk about variance and extra probabilistic info to be gained)

(maps of 4-label full predictions)

(maps of all-label full predictions)

5.4 Multi-Output Predictions

Looking at the Logistic Regression map from Figure 5.10 (as the rest provide little to no useful information), 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 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.

5.4.1 Coercion of Common Regression Machine Learning Algorithms

To do this, we first look at the regression version of algorithms used in the previous section, instead applied individually to each of the label distribution values in the original dataset, rather than simplifying them down to a single habitat label.

Multi-output average Errors

Algorithm	Average Error	Labels Used	Average Row Sum*	Min Row Sum	Max Row Sum
SVR	0.2073	4	0.7407	0.1687	1.4418
LinearRegression	0.1827	4	0.5224	0.1136	0.8421
KNeighborsRegressor	0.1698	4	0.5365	0.0	1.0
RandomForestRegressor	0.1722	4	0.5267	0.0	1.4
SVR	0.0983	24	1.8567	1.826	1.8834
LinearRegression	0.0463	24	0.155	-0.1624	0.4006
KNeighborsRegressor	0.0434	24	0.1655	0.0	1.0
RandomForestRegressor	0.0441	24	0.1842	0.0	1.3

TABLE 5.3: Average errors of multi-output versions of single-output regression methods

Note that the smaller errors when predicting using all 24-labels do not mean that it is 'better' to do so - because the values in the training set are smaller to begin with, it follows that the variance in predictions will also be smaller as a result. To visualise how these results translate when each of the models' trained parameters are used to predict the labels over the query space, heatmaps of each label were used, to allow easy identification of information such as which areas are particularly domianted by a single habitat, where they are particularly scarce, or areas where multiple habitats co-exist. Full predictions using SVM Regression was ommitted largely due to its poor performance observed from the average error above, in addition to the extensive computational time required for predictions compared to the other three algorithms.

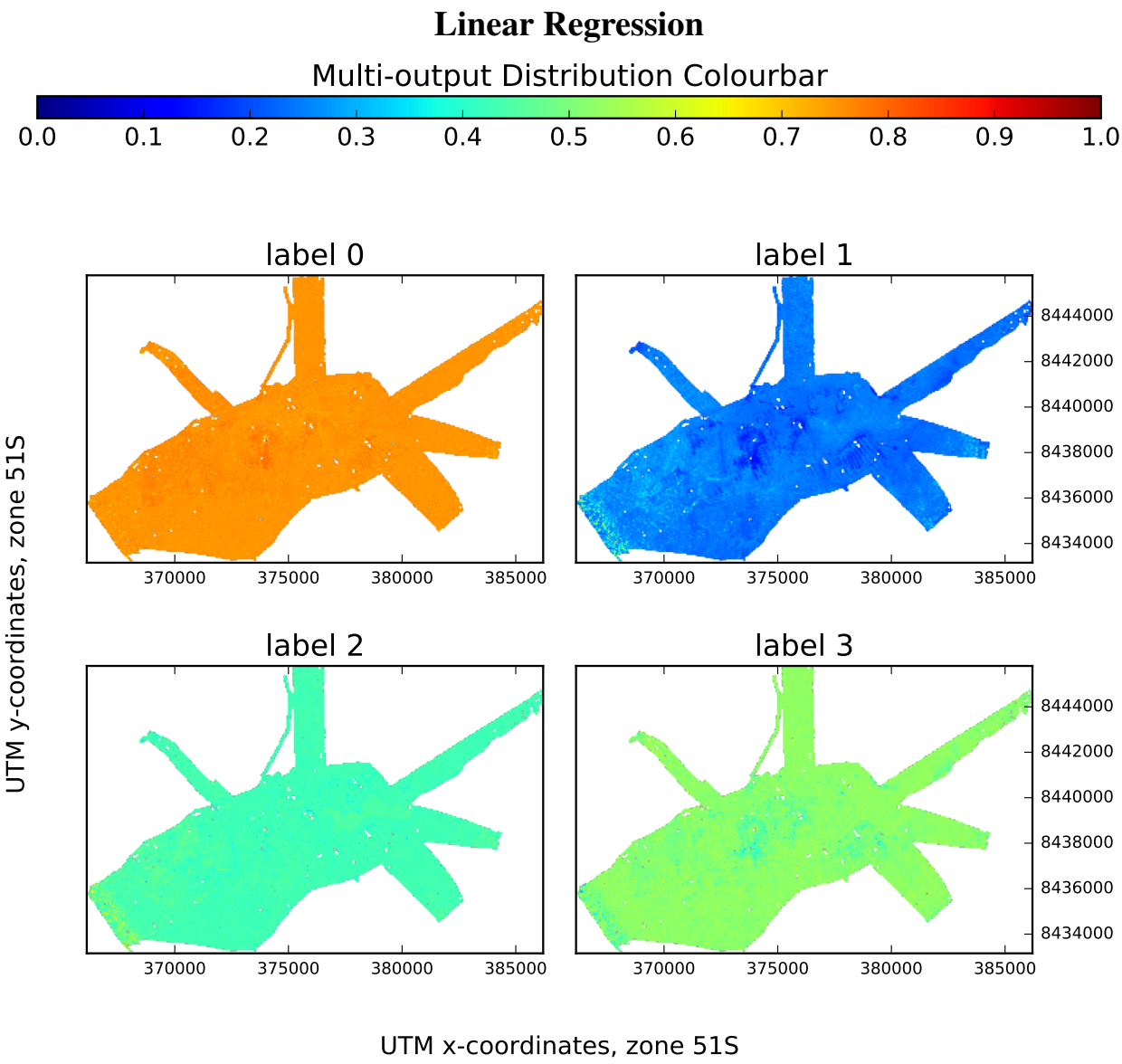


FIGURE 5.11: Map of full query Linear Regression 4-label predictions

It is immediately apparent that while linear regression has an average error within reason, the maps generated are likely faulty - each label's distribution is almost consistent throughout the 200km-squared area, a uniformity that is highly improbable if not impossible, given the large area of benthos covered. Moreover, the sum of the normalised predictive distributions at most points sum to almost 2 - well beyond the expected value, 1.

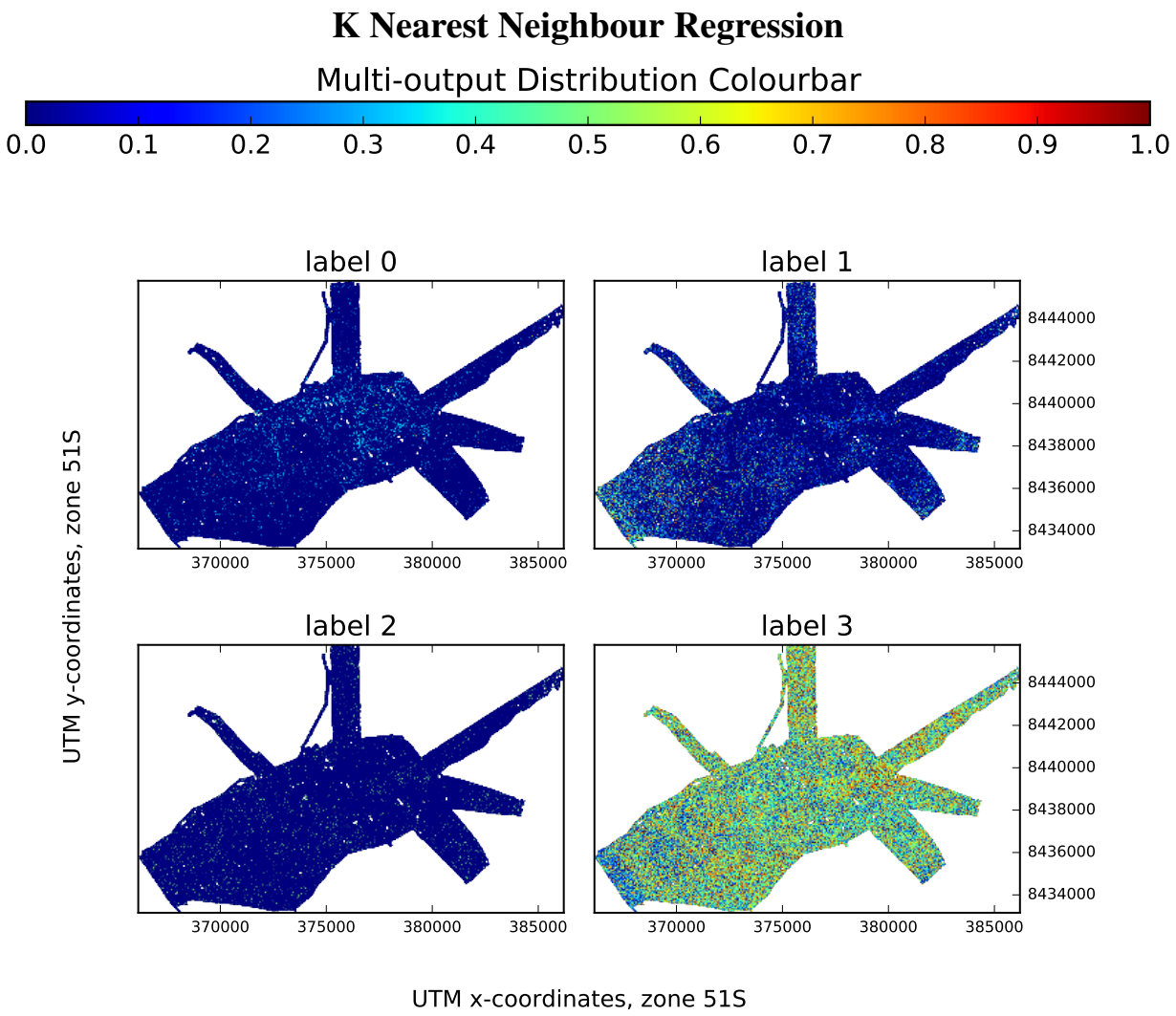


FIGURE 5.12: Map of full query K Nearest Neighbour Regression 4-label predictions

K-Nearest Neighbour performs slightly better, as there are actually visible differences in the distribution of label 3 throughout the area, whilst showing traces of the other three labels. However, the failure for label distributions per point can be easily observed here, considering that a sizable portion of label 3 areas occur at roughtly a 0.5 rate, but the combination of other labels in the same space quite clearly don't make up the remaining 0.5, as they mostly sit near the 0.0 mark. Moreover, the predictions themselves are very noisy - there are no observable contiguous areas where the occurrence of a habitat occurs at a similar rate, and nor are there visible smooth transitions where a habitat goes from high to low density.

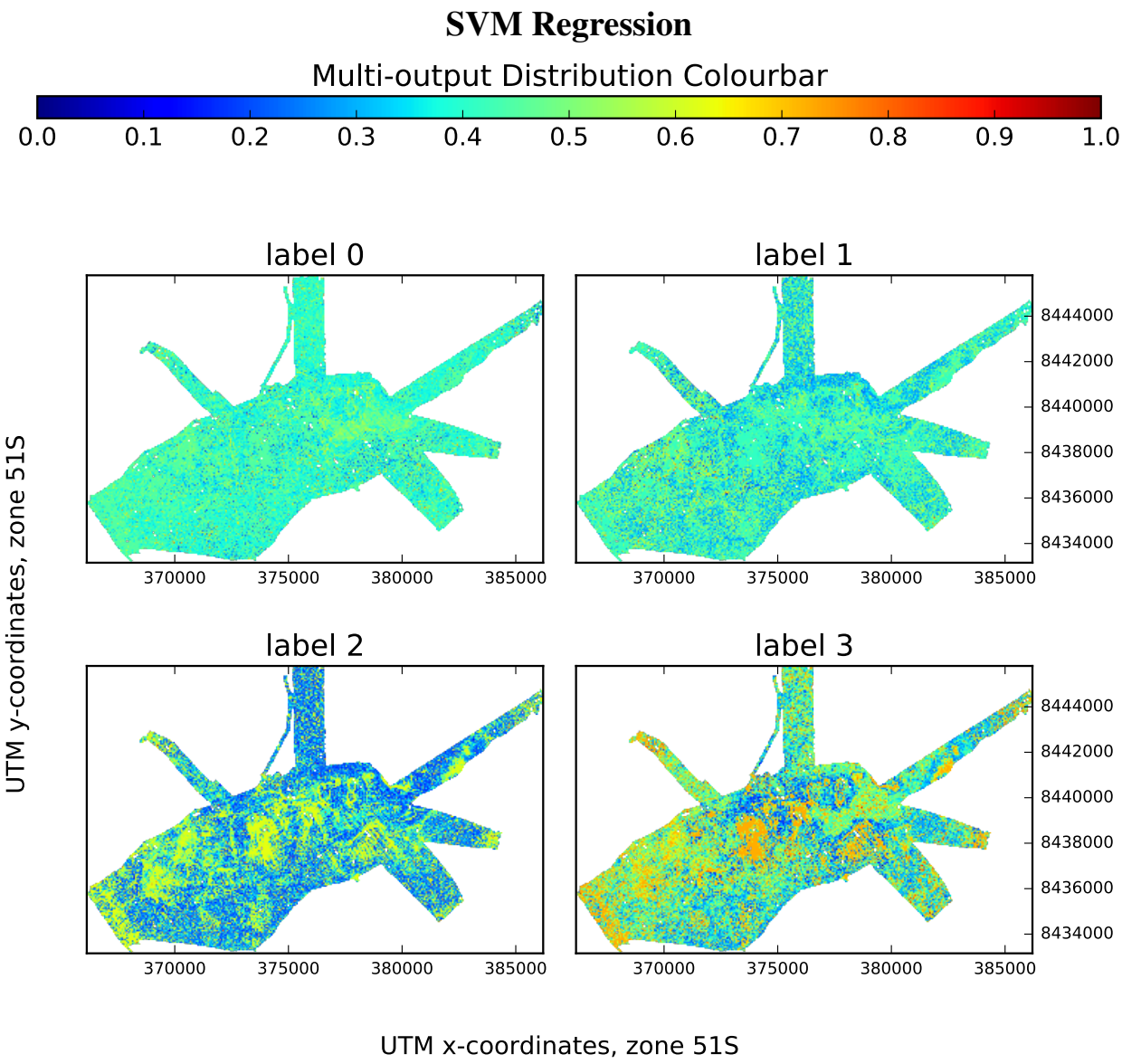


FIGURE 5.13: Map of full query SVM Regression 4-label predictions

SVM Regression is able to provide more reasonable predictions showing distinct areas of specific habitats that are more 'natural' compared to Logistic Regression and K-NN regression thus far. However, considering the rarity of labels 0 and 1 in the original data, it is quite unrealistic that they could be almost uniformly present throughout the entire of Scott Reef at such a high rate. Again, visual inspection indicates that a large portion of the map has violated the constraints of the normalised label distributions summing back up to 1 at each datapoint.

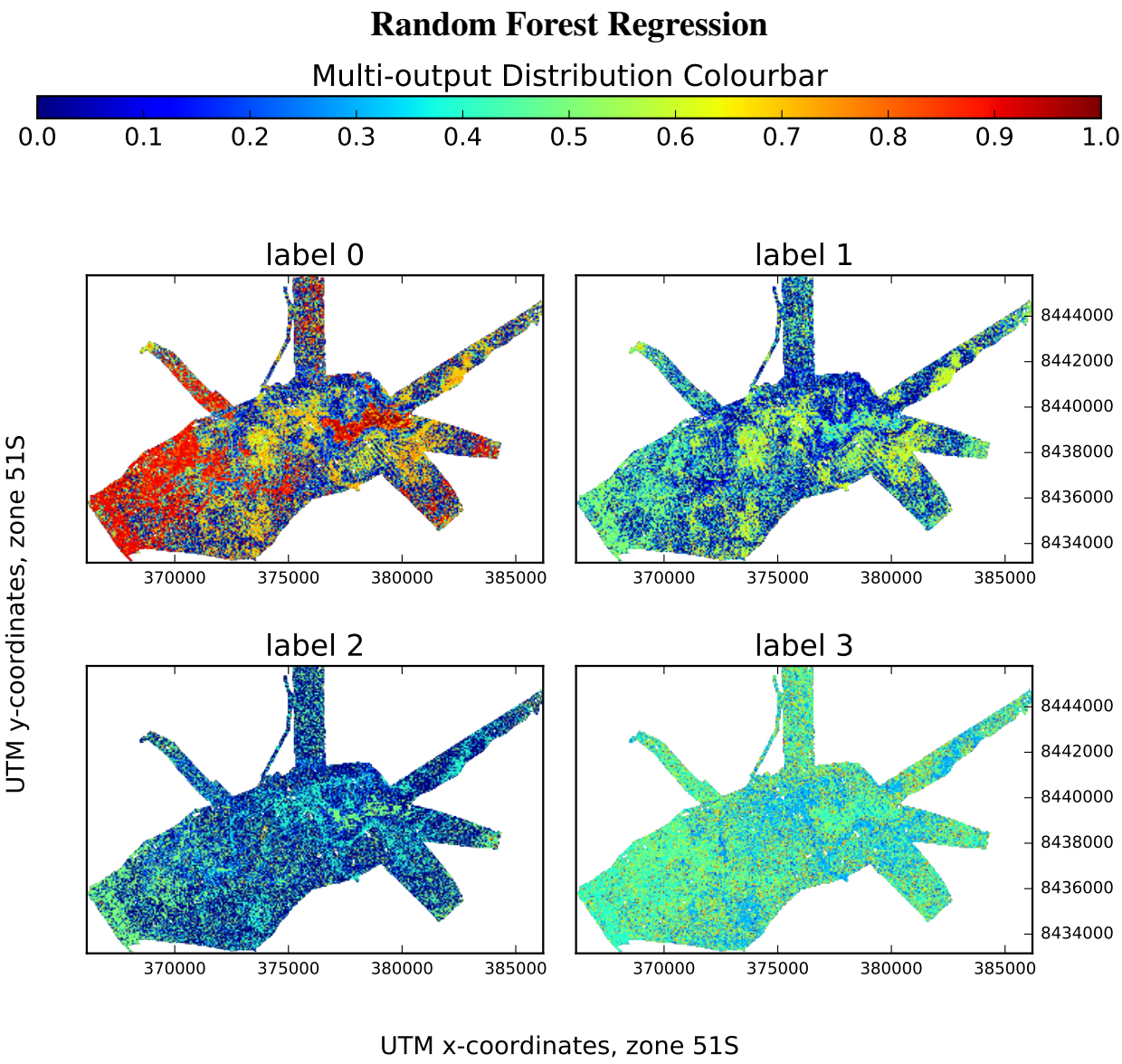


FIGURE 5.14: Map of full query Random Forest Regression 4-label predictions

Random Forest Regression produces a similarly varied map, but very different to that of KNN and SVM regression in some key areas. Labels 1, 2, and 3 have a notably higher predicted occurence rate throughout Scott Reef, with visible swaths that outweigh label 4 in the region. Without marine biologist expert or similar input, however, it is hard to determine whether the large regions of the less common habitats predicted by the Random Forest Regressor are likely in a location such as Scott Reef.

As a result of the constraint requiring label distributions per point to sum to 1, the above use of independent regressors is more of an illustrative exercise than one that can be relied on for real-world use - even if some of the information it provides appears to be of some use. In the extreme cases, some coordinates had no labels at all (the distribtions 'summed' to 0), while others were as high as 3. These suffer from the same disadvantage as in the previous section on single-output predictions by stating predictions in absolutes rather than providing confidence levels - and again, we apply Gaussian Processes to the problem to attempt to alleviate this, but using regression this time around.

5.4.2 Coercion of Gaussian Process Regression

Although a Gaussian Process is only designed to deal with single outputs, each of the label distributions per datapoint are seperate values, meaning it is possible to use multiple Gaussian Processes to allow it to work as a multi-output model. Morever, compared to the previous coercion of deterministic methods, we can use the variance of the Gaussian Process over each label to visualise the uncertainty of predictions. In contrast to previous methods however, due to the established $O(n^3)$ complexity, it was impractical to perform 10-fold cross validation on the full ≥ 5000 data points in the downsampled training set. Instead, a method similar to argmax, 230 points per class, 920 total, randomly sample the 920 until predictions on remaining points gives best performance was used to select the training points.

Labels used	Root Mean Squared Error
4	0.14409
24	?

TABLE 5.4: Multiple Gaussian Process Regression average error for the two label sets

Figure 5.15 below shows the full predictions for the 4-label dataset, with the majority of predictions for labels 0, 1, and 2 lying in the range $[0.1, 0.2]$. This is a likely indicator that the model and its parameters are inappropriate for the problem at hand, despite the variability of label 3 providing a more reasonable prediction of its occurrence. Looking at the actual predicted

values, however, reveals how far they have deviated from summing to 1 per point - waiting for another prediction to finish to pull values back up.

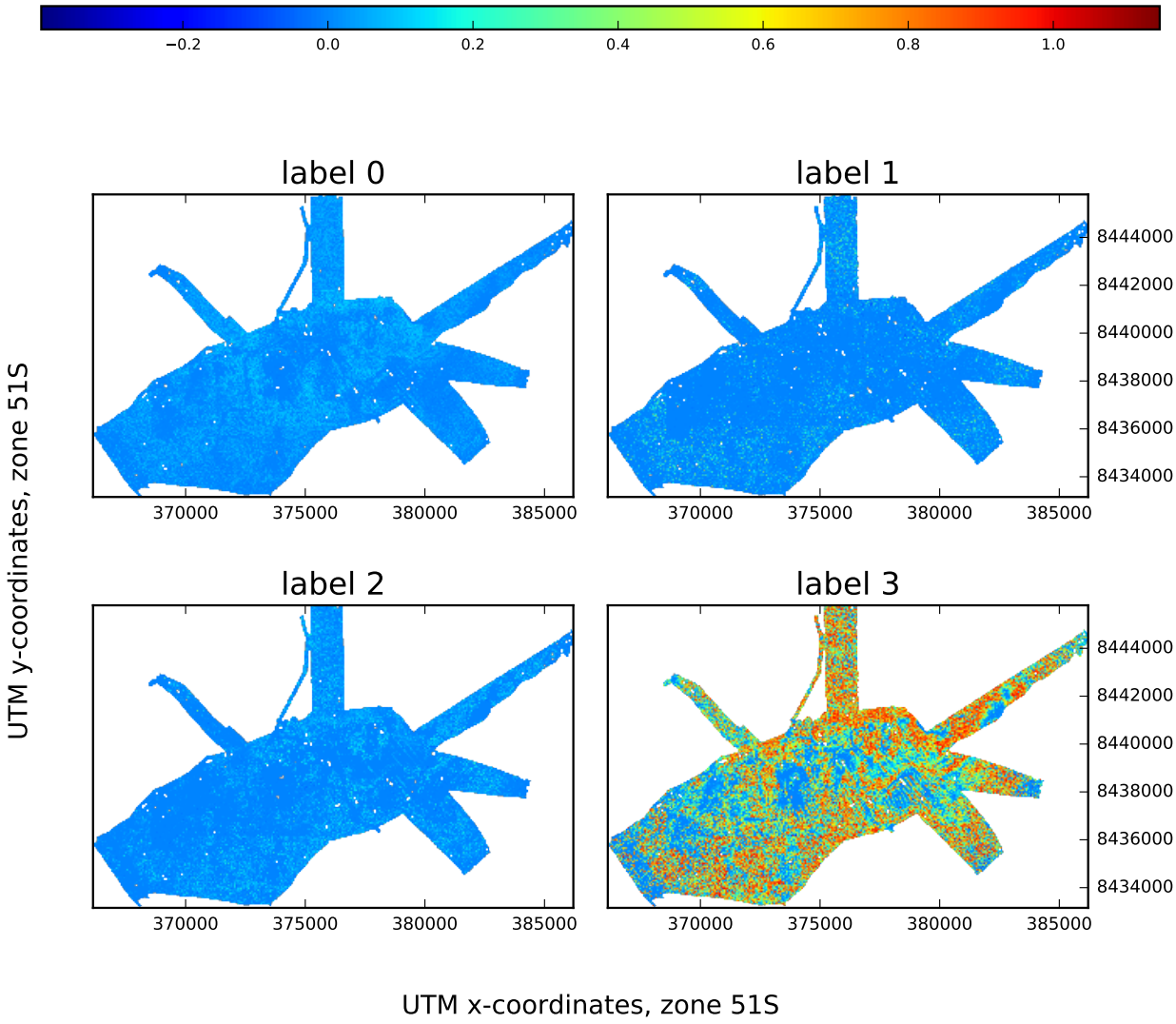


FIGURE 5.15: Gaussian Process predictions on full query data

Since multiple Gaussian Process Classification (Regression) does not discard a majority of the data compared to single-label Gaussian Process Classification, observing the variance is now more meaningful as *each* each of the possible c classes has a probability and corresponding variance of occurrence. However, as we see in Figure 5.16, not much can be gained from the

4-label case, considering that labels 0, 1, 2 have almost no variance, with only label 3 contaiing any visible sections with variance higher than 0.13, going as high as 0.41. Given that none of the methods so far are able to simultaneously (at least mostly) adhere to the constraints of the Dirichlet Distribution without explicitly incorporating it or provide realistic predictions given the rich habitat distribution data, we turn to a model that can in theory do both.

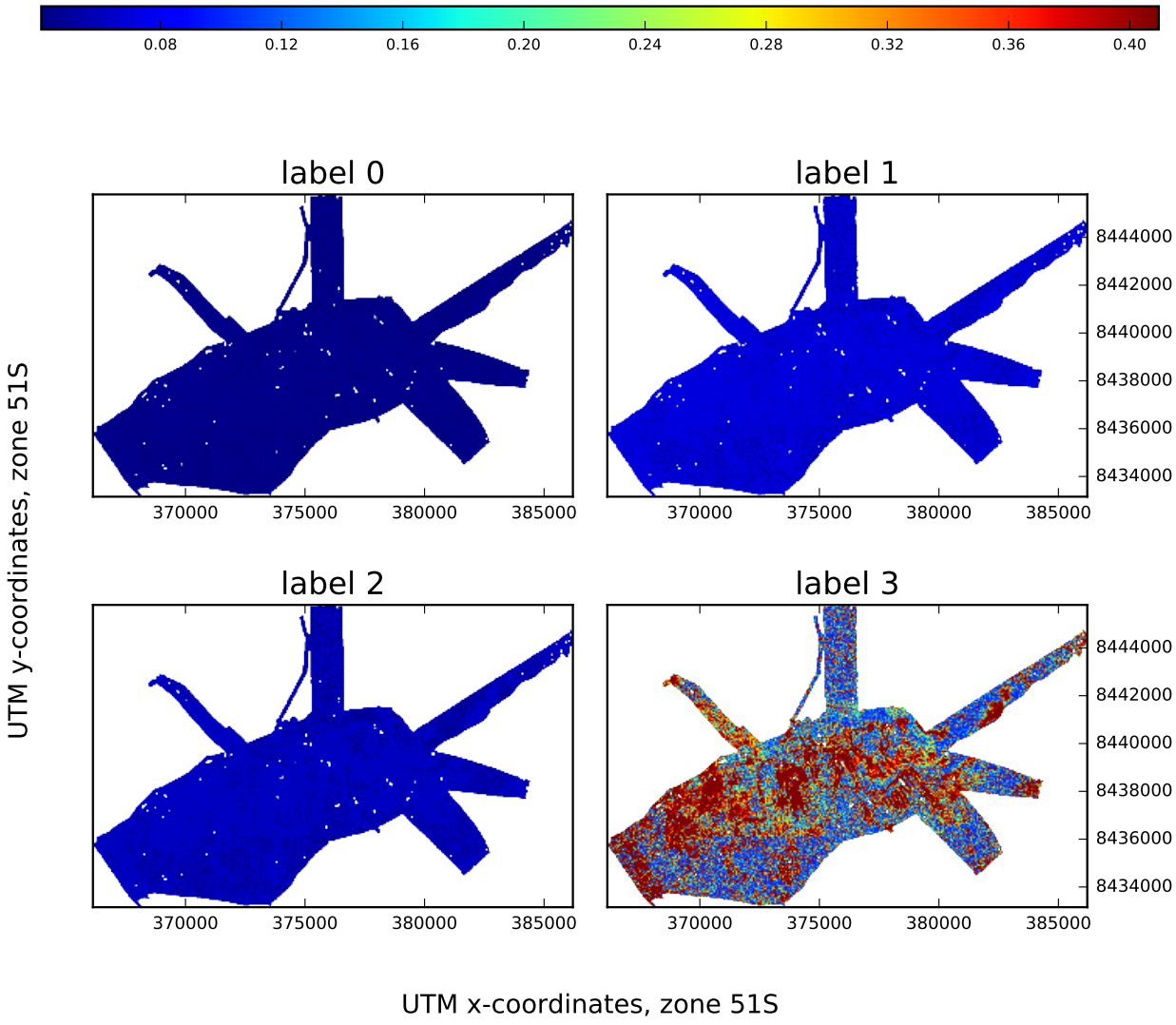


FIGURE 5.16: Gaussian Process variances on full query data

5.4.3 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.

Labels used	Root Mean Squared Error
4	0.17664
24	0.05916

TABLE 5.5: Dirichlet Multinomial Regression average error for the two label sets

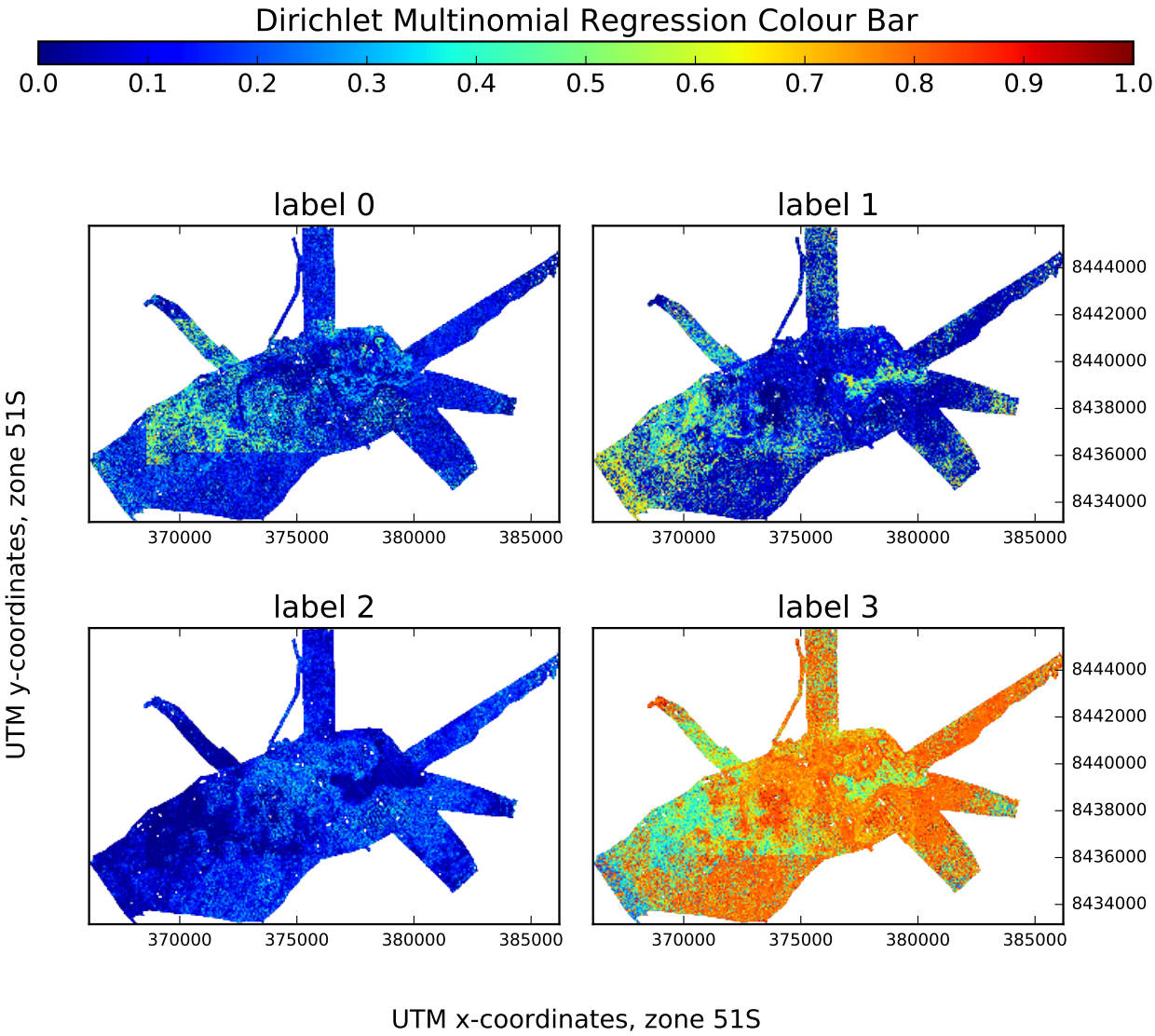


FIGURE 5.17: Distribution heatmaps over each label (in the simple 4-label case) for Dirichlet Multinomial Regressor output on query points

For the 4-label case, we can see some similarities with the models generated using the previous methods, with certain regions matching up to different predictions. For example, all predictions were able to agree on the significant presence of label 3 in the arm of the reef in the direction of the upper right corner of the region, though its dominance was in contention - with the Dirichlet Multinomial claiming parts of this area were around 0.75 label 3, whereas Linear, SVM, and kNN regression all predicted it to be mainly around 0.5. Now that we have correctly modeled

the data (at least in terms of properties of the predictions), more detailed observations can actually be made without the validity of the predictions being brought into question. This was the only model where predictions for label 3 exceeded a 0.5% occupancy rate, with roughly $10,000$ points with over a proprtion of 0.9 of label 3. 82% of points contained over 0.5 label 3, and 30% over 0.8. These statistics are attained with a single line of Python code (that runs essentially instantaneously) once the predictions have been made. Being able to conveniently determine these properties of the benthic body can be of immense use - if new low resolution bathymetry data and images is collected every several years, the rate at which the occupancy of certain habitats can be calculated without the need for expensive and possibly time-consuming post-processing of predictive maps.

The Dirichlet Multinomial also allows us to observe the variance for each label at each data-point, which can be interpreted as the 'confidence' in which the model has that label l at data point x occupies some ratio r of the points at that particular coordinate. This is invaluable information for conservation, for example, as it attaches a level of risk or assurance to any decisions or actions that need to be carried out, in contrast to the uncertainty when dealing with absolutes such as with linear regression in Figure 5.10. These confidence levels can be ovserved in Figure 5.18, where the variance over the query space for each label is mapped on a separate heatmap. The trend that can be seen here is that the model has generally high confidence (low variance) in the areas there are few to no labels, whereas the areas with a high occurrence of a particular label generally results in higher variance as well.

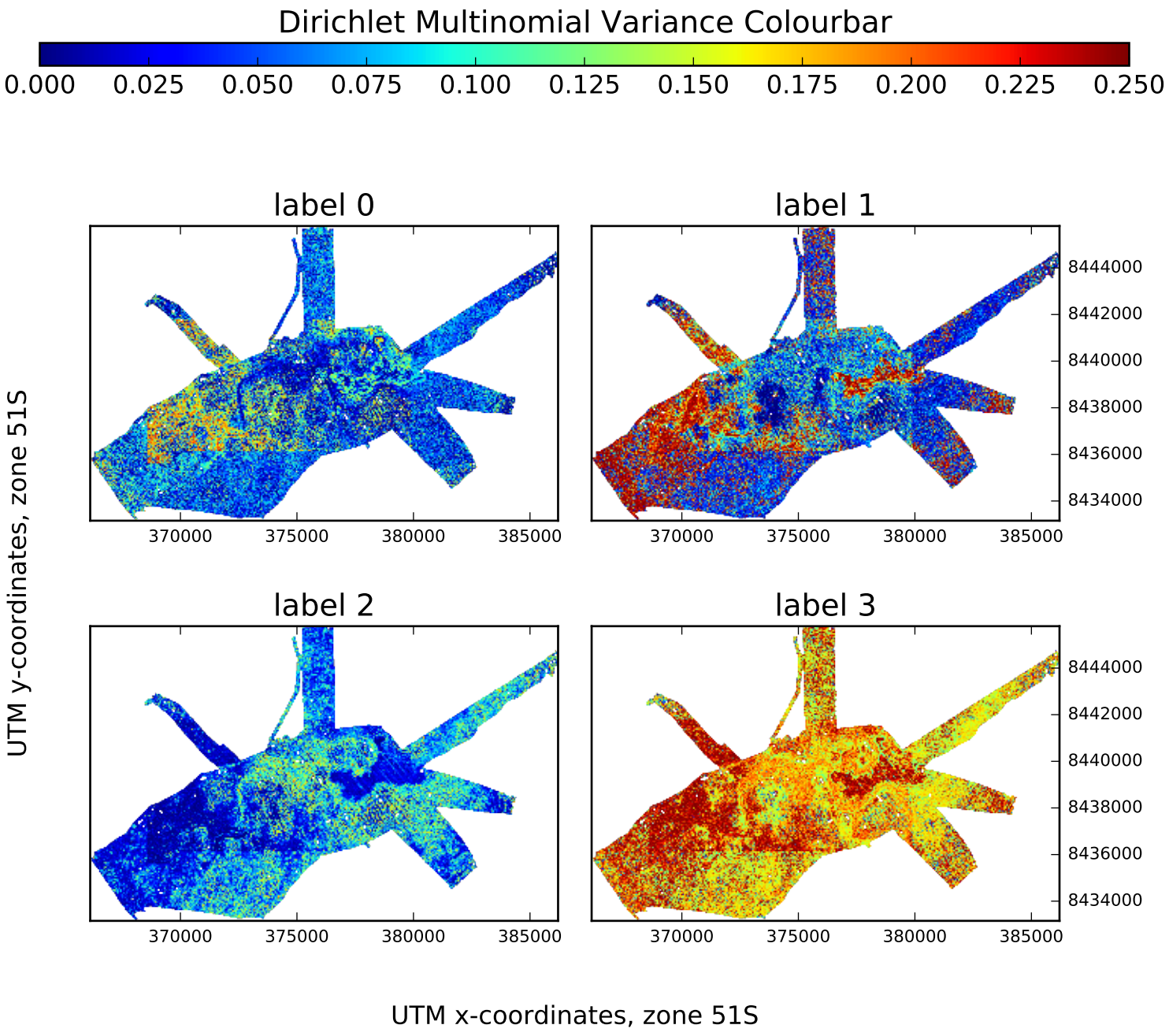


FIGURE 5.18: Variance heatmaps over each label (in the simple 4-label case) for Dirichlet Multinomial Regressor output on query points

5.4.4 Dirichlet Multinomial Predictive Map Variance

While the basic results for the Dirichlet Multinomial were obtained using the Maximum a Posteriori (MAP) estimate, 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 converged, then create a large number of maps from these chains (that were actually the weight parameters) to visualise which areas of the predictive map would remain relatively consistent across chains, and which ones wouldn't. Both the 4-label case and 24-label case were run for about a day, corresponding to a $4 * 19 = 76$ and $24 * 19 = 456$ dimensional space respectively in which the MCMC had to traverse.

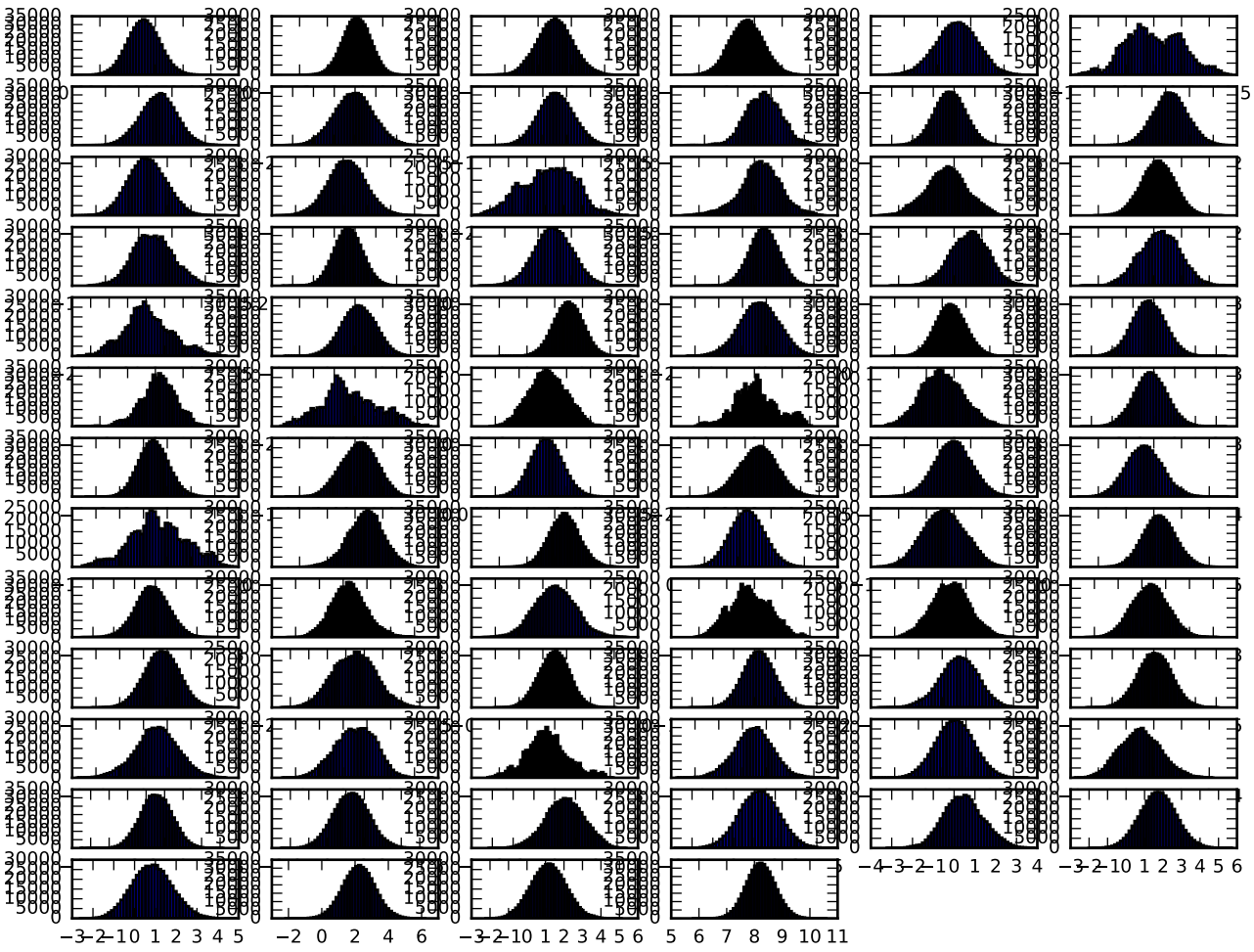


FIGURE 5.19: MCMC weights for 4-label, 19-dimension data case (need to separate this into separate images, possibly remove axis ticks)

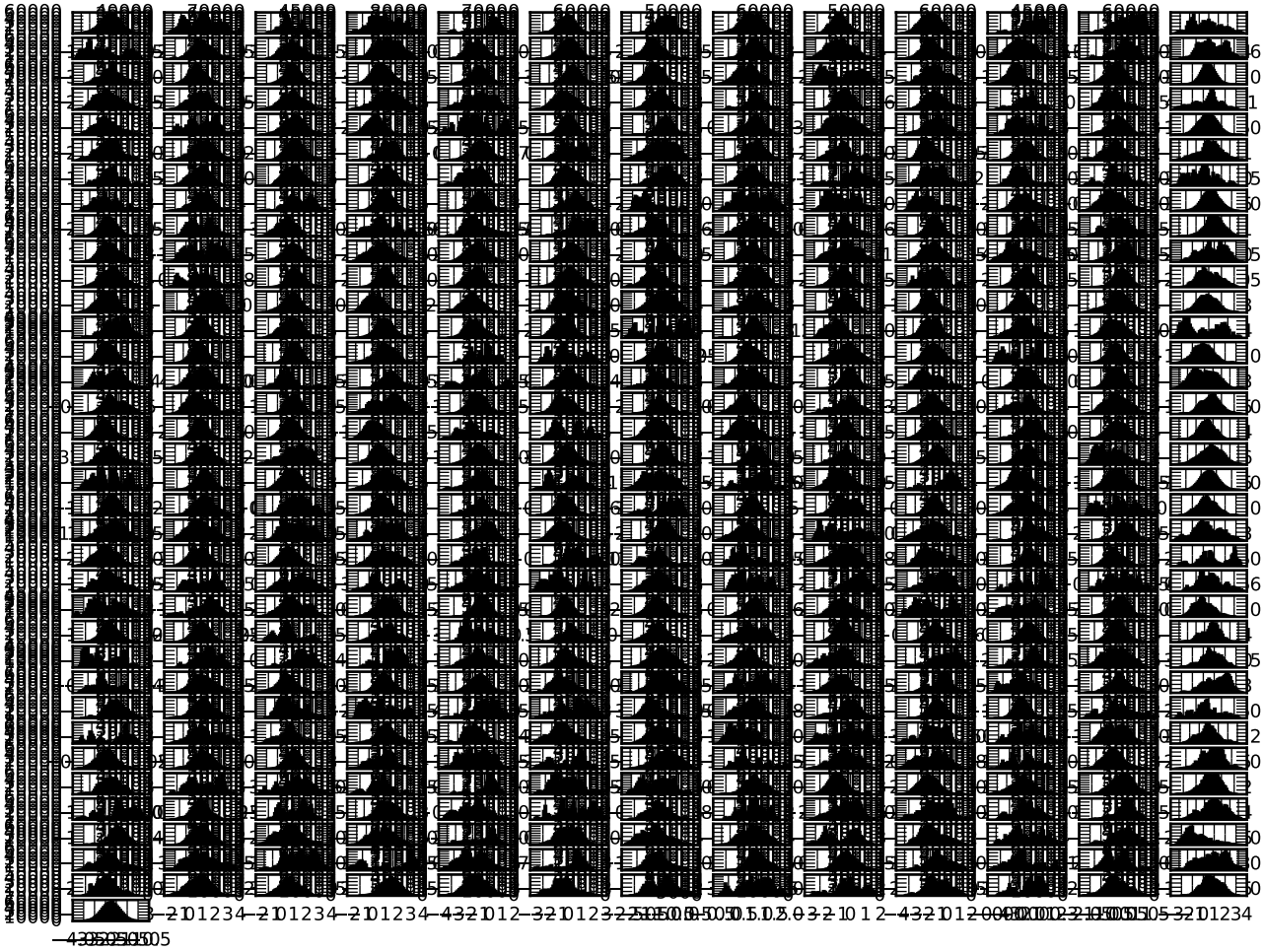


FIGURE 5.20: MCMC weights for 4-label, 19-dimension data case (need to separate this into separate images, possibly remove axis ticks)

5.5 Biodiversity

Another beneficial aspect of Dirichlet Multinomial Regression it inherently provides information about the distribution of different habitats in a given region, allowing observations on biodiversity to be made without requiring extra post-processing steps such as clustering, which

can be prohibitively expensive on datasets with millions of datapoints and tens (or more) of dimensions. To locate certain co-existence of species, the only step required would be to search over the space of predictions for the desired distributions simultaneously. As an alternative, or extra step, the predictive variance at these points indicate the confidence by the DM about the predicted mix of labels.

In contrast to the GP above where the uncertainty was greatest when there were even distributions of labels (don't know this yet! waiting for predictions to run), it is expected that the DM would be comparitively more confident that an even mix of labels exist in these areas. To obtain a sufficiently large area/number of points where two of the simplified four labels had a fairly even occurrence rate (with the other two labels only having trace amounts, if at all), pairs of labels were repeatedly sampled with the variable condition that both their distributions lie within a certain range (for example, $[0.4, 0.5]$, or $[0.2, 0.3]$), until a segment was found where the average variance over these points were significantly lower than the variance in label distributions across the overall predictions. The variance in this regions were then compared to that of a Gaussian Processes'.

Because the 4-label case already aggregated similar classes from the original 24 down to 4, there was limited 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 perform initial measures of biodiversity, areas where 2 or more labels co-existed were searched, but where their occurrence ratios lie within a certain range. The examples we show below limited this range to 0.03 - however, this is arbitrary, and also tunable depending on the specifics of the benthic region, as well as the goals of obtaining biodiversity metrics. These searches were automated over the query space to find sections where the average variance was noticably lower than other parts of Scott Reef, indicating that the model had a higher confidence of a particular mix of habitats in the given section.

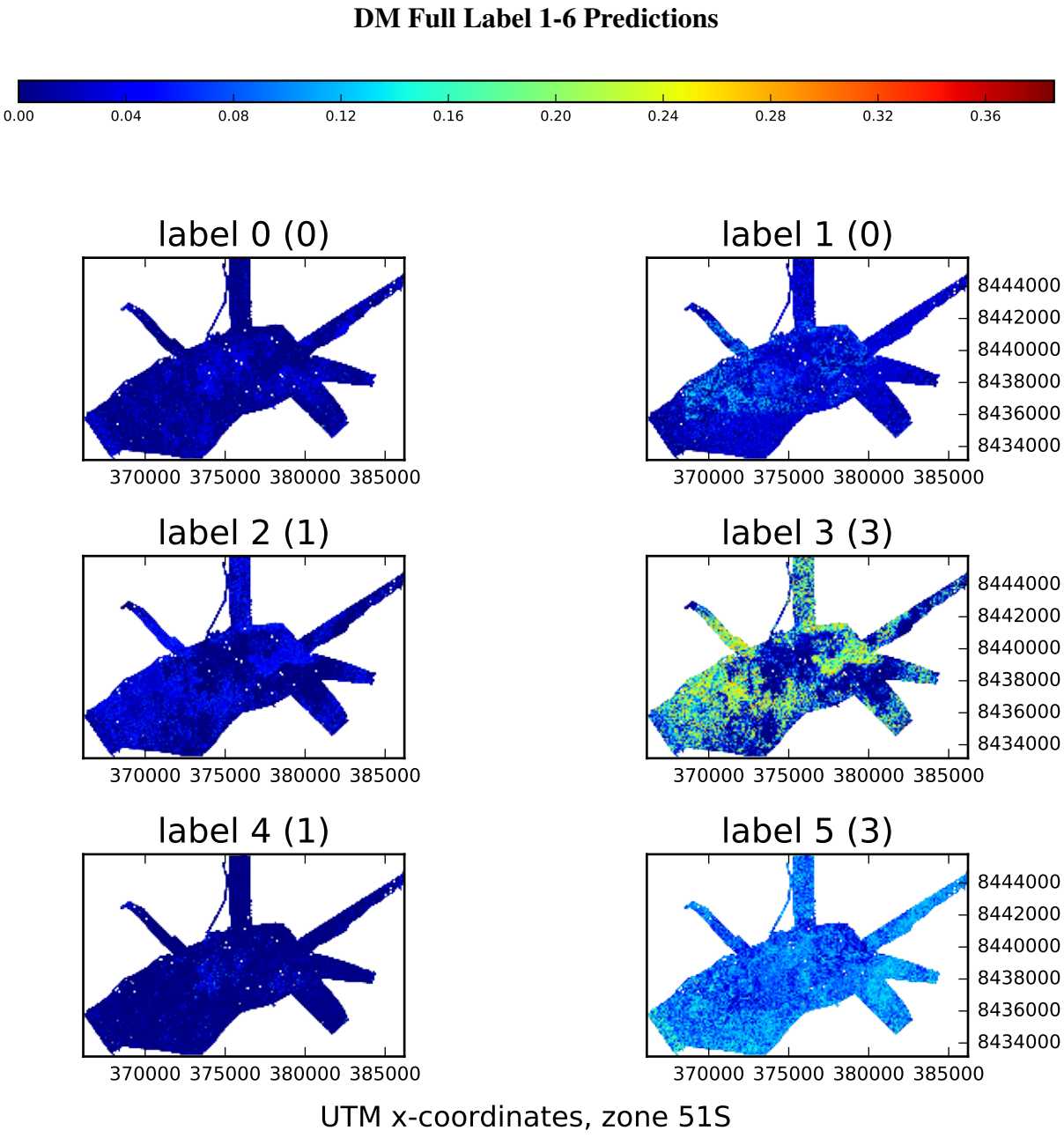


FIGURE 5.21: Distribution heatmaps over labels 1-6 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

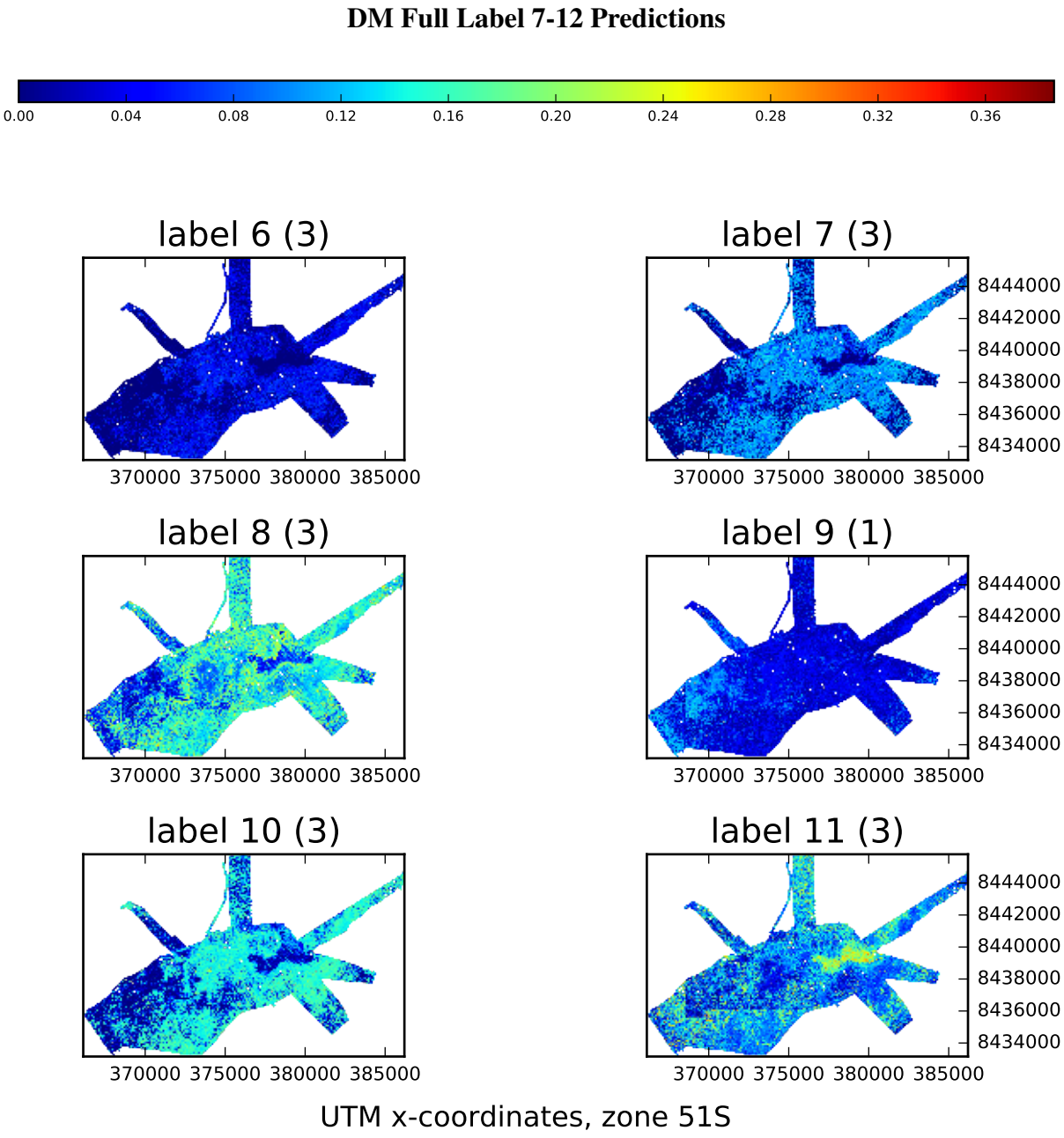


FIGURE 5.22: Distribution heatmaps over labels 7-12 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

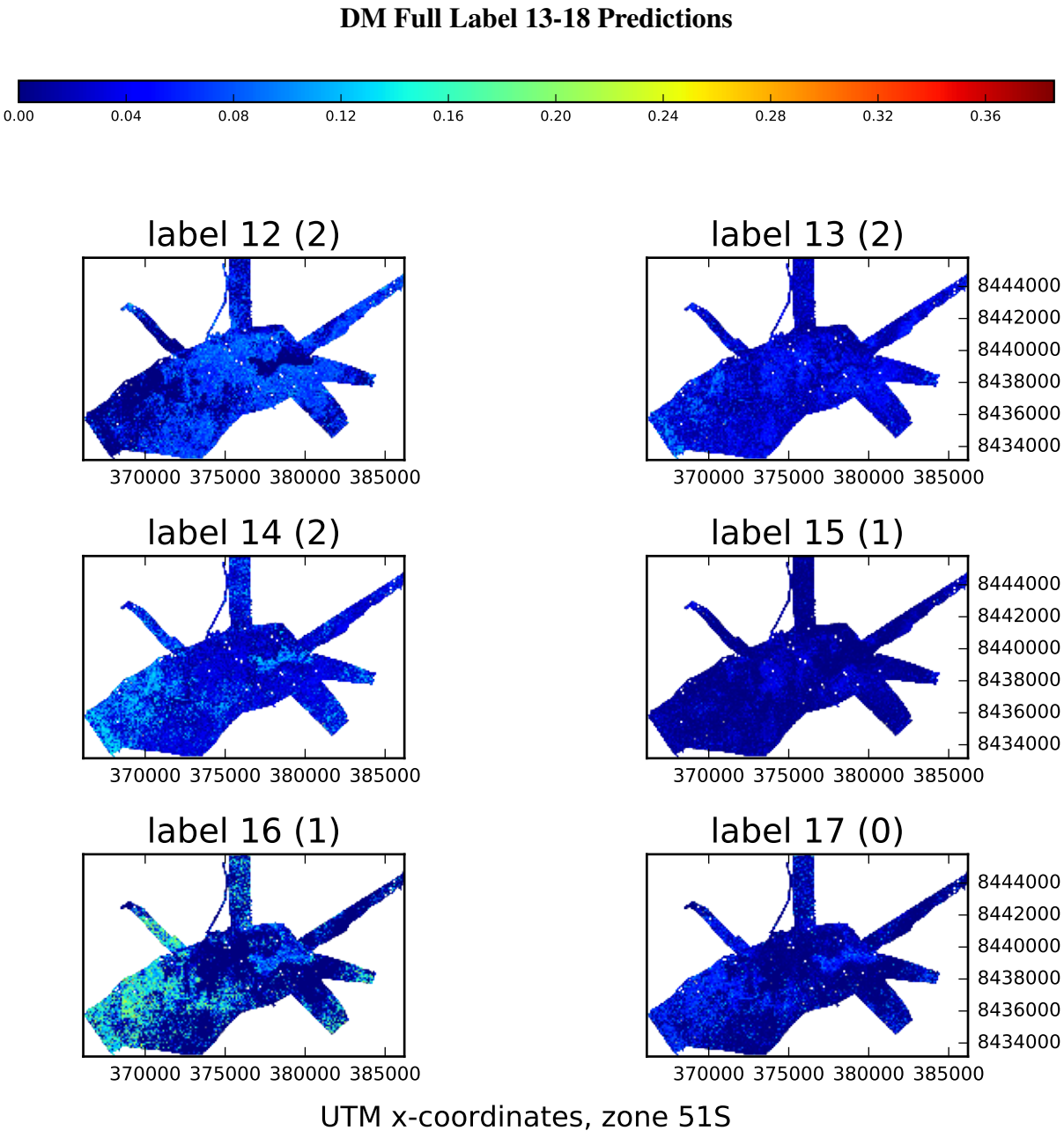


FIGURE 5.23: Distribution heatmaps over labels 13-18 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

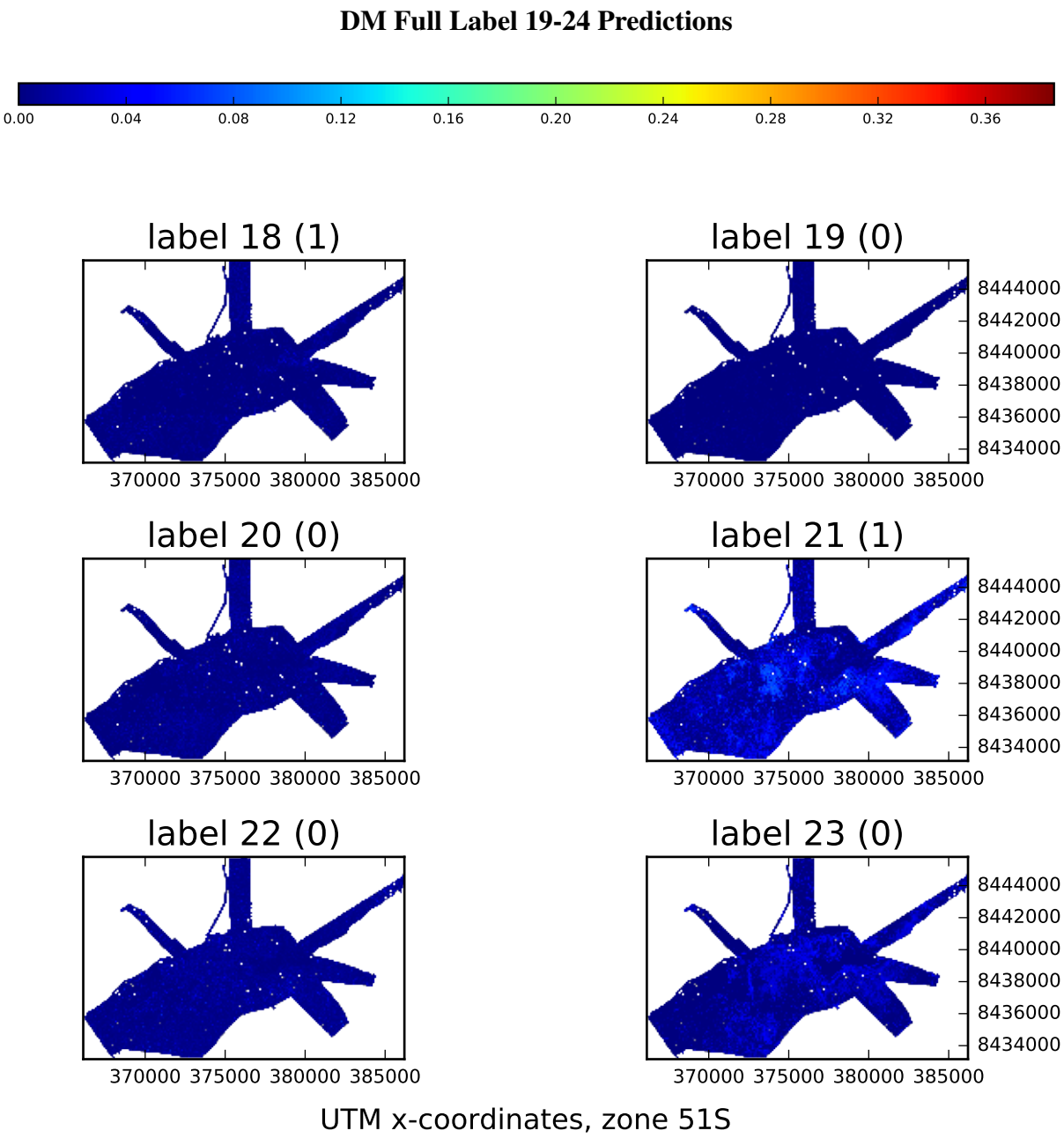


FIGURE 5.24: Distribution heatmaps over labels 19-24 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

[PLACEHOLDER] We can visually observe that labels $7, 8, 10$ (variants of sand) occur in the same regions, with the following occupancy rates, beyond areas containing only negliglble traces ($\equiv 0.00 - 0.02$) 7 at a ≈ 0.12 occpuance rate, 8 at $\approx 0.18 - 0.20$, and 10 at $\approx .17$.

What becomes apparent is that in the areas where the DM is confident of a mix of certain set of predominant labels, the GP 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.

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

CHAPTER 7

Conclusion

The conclusion goes here.

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

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.

Christina L. Belanger, David Jablonski, Kaustuv Roy, Sarah K. Berke, Andrew Z. Krug, and James W. Valentine. 2012. Global environment predictors of benthic marine biogeographic structure. *Proceedings of the National Academy of Sciences*, 109.

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.

Chang, Chih-Chung, and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27.

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

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.

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.

Braveheart Marine. 2016. Hydrographic services. `http://www.braveheartmarine.com/20-survey-dept/59-hydrographic-services`.

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.

addtocontentstoc

APPENDIX A

Appendix

---

things