Multi-output and Probabilistic Large Scale Benthic Habitat Mapping

JUSTIN TING

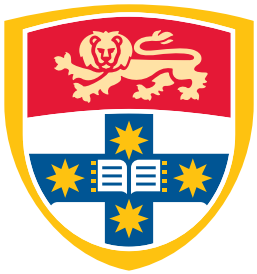
SID: 430203826

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

19 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 4 and chapter 5, where we look at the mathematical background

behind Gaussian Processes and Dirichlet Multinomial Regression. In chapter 6, 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

Notes

(1) when talking about LR/RF, include some basic maths

(2) refer back to these equations/etc. in later chapters when describing advantages/etc.

2.1 Overview

The process of benthic habitat mapping involves three key steps that the large majority of all studies in the area go through.[1]. In this section, we will give a brief overview of each of these steps, along with common procedures used in them across studies in this area.

(1) **Habitat Characterisation** - extracting properties of the environment such as rugosity (roughness), aspect (direction of slope), depth

(2) **Habitat Classification** - grouping the raw information about the environment into categories, such as sand, granite, etc.

(3) **Habitat Mapping** - using classifications with the larger scale bathymetry data to extrapolate habitat maps

2.1.1 Habitat Characterisation

not just resolution of data but modality. images vs bathymetry If we were able to collect high resolution data for the entire ocean's benthos - the job of creating benthic habitats for any given

[1] Ozcoasts.gov.au. (2016). Benthic habitat mapping: Mapping Overview. [online] Available at: http://www.ozcoasts.gov.au/geom_geol/toolkit/mapoverview.jsp

area would be (relatively) trivial. As this is prohibitively expensive, we instead collect large amounts of low resolution data, and small samples of high resolution data (between which we model a relationship). This subsection provides a brief summary of data collected and methods used to do so.

Remote-sensing data. Due to the cost of sea expeditions, it is economically infeasible to have marine vehicles (autonomous or otherwise) explore the entire ocean floor to confirm the ecological properties of all of Earth's benthos. However, we do need to collect sufficiently detailed data of large areas at a time, partiulcarly those of being mapped, and for this, remote-sensing data is used. These usually come in the form of acoustic backscatter data that involes the firing of sound waves towards the benthos, whereby their frequency and strength upon returning is used to deduce the depth of a particular material, as well the density of said material (from which a guess at the actual substance can be made - e.g. sand, mud, etc.).

Multibeam echosounders (MBES) are becoming a more frequently used method of collecting acoustic backscatter data (Calvert, Strong, McGonigle, and Quinn, 2015) despite older methods involving single beam echo sounders (SBES) being cheaper and easier to segment. This stems from the fact that the reduced cost comes at the expense of (potentially) accuracy, as well as lower resolution data. This is due to SBES' beam angle, i.e. the angle formed by the 2D flattening of the 'cone' shape of the emitted beams, ranging from 15-25°, whereas MBES' is 0.5-3°, depending on the particular system (Brown, Smith, and Lawton, 2011). The difference in angle means that data returned via SBES devices are more 'coarse', reprsenting less accuracy and granularity, whereas that of MBES is more detailed and can present more information. However, there is overhead associated with use of MBES, in that the considerably decreased angles means much more 'overalpping' data, adding complexity to the segmentation process.

Truthing Data. explain redundancy here, unclear what it's referring to - why is there redundancy? The most common methods to be able to obtain a sufficiently large truthing data set (but still trivially small compared to the area covered by remote-sensing data) are videos or images - though the former still requires post-processing to extract the needed images. The advantage that can be provided here, however, is the redundancy in data points (Rattray, Ierodiaconou,

J. Monk, and Kennedy, 2014) - but there is extra cost in time required to convert videos into the needed images (pre-proessing before feeding into algorithms for habitat mapping), an area that is in itself worth of research within the field. (Lucieera, Hilla, Barretta, and Nichol, 2013)

Other data. (why is water column correction important when correlating images with seagrass standing crop?) Other data that is less common, but also used to map habitats, is patterns in the water movement (such as tidal currents, wave action) (Brown, Smith, and Lawton, 2011) in the column of water above the area of benthos being mapped - a feature that provided useful input in arriving at an accurate benthic habitat map (in addition to sediment analysis). (Snelgrove, 1994) Other sources such as UNESCO have also verified the importance and significance of using water column correction techniques to obtain more accurate habitat maps, particularly when correlating images with segrass standing crop. [2]

2.2 Habitat Classification

more in-depth focus here, what kind of supervised/unsupervised ML algorithms are used for classification? Almost all studies use *in situ* '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). The two overarching ways to perform this classification are in the form of supervised and unsupervised algorithms.

Studies have used both supervised and unsupervised methods in clustering the initial data for the training step. 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

[2] Unesco.org. (2016). Water column correction techniques. [online] Available at: http://www.unesco.org/csi/pub/source/rs10.htm

are first used to create these clusters, after which an expert may be brought in to verify/simplify (or otherwise) the resulting clusters. (Steinberg, Friedman, Pizarro, Williams, and S.B., 2011)

2.3 Map Creation

The final step is map creation, which many papers related to benthic habitat mapping focus on - and also where the most variation occurs in terms of the method used. The various approaches used can be categorised into two broad categories. The first is a top down approach whereby the classification of the habitat characterisation data is validated (or otherwise) with the truthing data, and the second is a bottom up approach where the characterisation data is similarly clustered into classes, but not to directly represent a particular habitat - instead, the aim is to find a relationship between the acoustic data clusters and the truthing data clusters which we can model. Using this model, we can then extrapolate the acoustic data which doesn't have corresponding truthing data to create the habitat map. (Ahsan et al., 2011) We will explore this aspect more when looking at how the mapping process has evolved over time and the improvements that it has brought about.

2.4 Non-Machine Learning Approaches

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 drawing broad conclusions about the effectiveness of a collection of tools in creating habiatat maps, deeper analysis is done on subsets of the data to attempt to clarify some of the complexities and intrinsic properties of benthic habitats and ecosystems themselves. 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 contirubted heavily to a higher taxonomic 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.

Certain organisations, government bodies/etc. will also provide guidelines outlining the classification proess. For example, the European Nature Information System website and the Australian Government's 'Interim Marine and Coastal Regionalisation for Australia'[3] both provide classification schemes for people creating habitat maps or other similar efforts.

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 fully utilise the information that we gather.

2.5 Machine Learning 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 papers can rely on collective findings of others to launch their own research and look further into particular lines of inquiry, start new ones altogether, or evaluate

[3]Unesco.org. (2016). Water column correction techniques. [online] Available at: http://www.unesco.org/csi/pub/source/rs10.htm

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 revisiting common machine learning techniques and their application in the various stages of benthic habitat mapping along with the benefits they provide.

2.5.1 Deterministic Machine Learning Algorithms

In this section, we will review some machine algorithms that can be used in benthic habitat mapping processes - whether that be in the initial clustering stages of (ideally) independently gathered datasets such as acoustic backscatter data and collections of high resolution images, or the actual classification of 'new' (or testing) data in determining their predicted habitat classes.

Multinomial Logistic Regression. Multiple Logistical Regression is one of the more basic machine learning algorithms that can be used to predict habitat classes, and falls under the 'supervised learning' category as we have the 'output' for the feature vector in the intial data. 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). This only applies to domains with two classes, however - to use this technique for classification where we have an unbounded (though usually still relatively low) number of classes, 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 maximum likelihood estimation (MLE), similarly to logistic regression. However, the difference is that whereas logistic regression only requiring a single logit function as its nominal variable is dichotomous, 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.

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.

Multi-class Support Vector Machines. Although support vector machines fell outside the top three overall supervised learning algorithms in terms of performance (accuracy), their non-parametricity could potentially be of benefit given that our knowledge of the complex relationships between elements of benthic habitats are limited. Moreover, despite SVMs being rarely used anywhere in the field, they are "acknowledged to be very competitive descriminative classifiers in machine learning literature" (Ahsan et al., 2011).

However, SVMs in their base form only support classification into two classes, requiring modification to the original algorithm to support more - an active area of research that has not found any single 'best' way to perform this algorithm extension yet. To do so, there are two, basic main approaches available, one being a **one vs. all** approach, and the other being a combination of all **one vs. one** approach. We can hence quite clearly see that given C possible classes, the first would require C separate classifiers, whereas the latter would require $\frac{C(C-1)}{2}$ classifiers (Murphy, 2012). Multi-class SVMs were used in Ahsan et al. (2011) for illustrative purposes, 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.

2.5.2 Probabilistic Methods

The classifications being made regarding benthic habitats naturally involve uncertainty, as we are still learning the relationship between differnt characteristics of benthos with the varying

communities of fauna and flora that reside there. Whilst guessing the most likely class for a particular domain deterministically has its practical applications, it is arguably more *natural* to represent the uncertainty (Rasmussen and Williams, 2006). As our understanding of marine environments is still quite weak (of the United Nations, 2004), it is debatable whether deterministic results are always appropriate when being used to make high level management decisions relating to marine environments. While deterministic methods will create a model that attempts to explicitly account for all variables, probabilistic models deal with joint distributions over all the variables. As we need to better understand "the complexities of coastal system functioning rather than simplfying and scaling down the system into smaller components" (Diaz et al., 2004), this feature can be esepcially valuable seeing as there is simply not enough 'expert knowledge' to adequately, explicitly model the relationship across a range of variables.

Illustrative Example. not an illustrative example - give some actual figures/graph of crossover point A simple example of this can be seen when comparing the deterministic approach of a logistic regression classifier, with the probabilistic Naive Bayes classifier. Starting from no data, up until a certain threshold, a Naive Bayes (NB) classifier will actually provide a more accurate classification as it approaches its comparatively higher asymptoptic quicker, after which point, one there is sufficient data, the logistic regressor will provide the better results (Ng and Jordan, 2002). In this simplified example, an analogy can be drawn where the data used up to the threshold when the NB classifier performs better represents a lack of knowledge about the data causing the logistic regressor to underperform, where as the continued addition of data represents more understanding (more data points) of the domain, allowing logistic regression to then outperform the NB classifier.

Gaussian Mixture Models. Gaussian mixture models (GMMs) are parametric models that "model the distribution of data as a set of clusters, where each cluster is a multivariate Gaussian" (Ahsan et al., 2011). In this particlar paper, GMMs are compared with classification trees, which it is found to perform better than in most cases, but were also predicted classes from unseen data with higher certainty than discriminative methods. This is because of its generative

nature that accounts for the distribution of bathymetric features, allowing it to model the joint distribution of the classes as well as features. Moreover, each (function = distribution)Gaussian function within the model has its own mean and covariance matrix, which also contributes to its powerful modeling ability. However, the use of GMM may have been hindered by the dimensionality of the data - while only five properties were measured, each was calculated for a varying number of scales for the input vector, meaning the 'features' were at least some multiple of five. As this exceeds the recommended six dimensions for use with GMM, application to a very large dataset may be beyond reasonable computational ability.[4]. To avoid this, the feature vector may have to be truncated to contain the bathymetric properties for only one particular scale at a time.

Using Gaussian Processes. A recent study used probabilistic methods to develop a mapping between the clustered acoustic data to continuous cluster probabilities, as opposed to discrete cluster labels, thus representing the certainty of the results obtained. Using Gaussian Processes which do not inherently support classification, Bender et al. (2012) extended the probabilistic least squares classifier to retain the information regarding certainty of class membership that exists during the classfication process, rather than discarding it in the traditional method. By evaluating the probabilistic results of PTLSC by comparing its results with the actual cluster probabilities obtained in the classification of the images via an unsupervised variational Dirichlet process model, it was shown that the PTLSC method performed better than a PLSC trained directly on the discrete cluster labels in terms of accuracy, mean squared error, and mean variance as well. This demonstrates that while both PTLSC and PLSC err in their predictions when dealing with the transition different boundaries, by maintaining probabilistic information in the PTLSC, it is able to make slightly better judgements in such cases.

Gaussian Processes and large datasets. However, Gaussian processes involve a matrix inversion process that requires an O(n^3) operation which does not scale well with large datasets. To overcome this whilst reaping the benefits of Gaussian processes, Bender et al. (2012) extracted subsets of the original dataset on which to perform analysis - a small, randomly chosen

[4]Nickgillian.com. (2016). GMM Classifier âĂŤ NickGillianWiki. [online] Available at: http://www.nickgillian.com/wiki/pmwiki.php/GRT/GMMClassifier

portion from three Gaussians, of the initial millions of observations. While this has still provided a high accuracy for all methods tested, there is likely information to be gained by being able to use a considerably larger portion of the dataset. To do this, a method would be required to generate sparse covariance matrices through approximations (Bickel and Levina, 2008), or use of functions that guarantee sparseness as a property (Melkumyan and Ramos, 2009) - something that can be explored in future work. To illustrate how the obstacle can be overcome, the latter paper describes a method whereby, rather than inverting the covariance matrix in its raw form, a threshold is calculated at which point, rather than observing the normal 'tapering' off of covariance values, they are simply set to zero beyond that point. This will result in a significant portion of the covariance matrix being populated with 0s, at which point inversion of the sparse matrix can be performed for which there are known efficient methods. However, there have been more ways of sparse approximation GPs that other studies have explored.

Sparse Approximation Gaussian Processes. m isn't clarified here, plus n<m is incorrect, should be m«n

lots of things here have become irrelevant (talks about 'optimal' methods that aren't implemented/included in experiments - limit to relevant ones, i.e. the GP ensemble methods

It is of importance that a number of methods of dealing with sparse approximation of GPs are taken into account if the aim is to deal with large GPs in the inversion step. (Quinonero-Candela and Rasmussen, 2005) explores exactly this, immediately discounting the "subset of data" (SoD) method as being non-competitive due to it not being able to represent the original data to a reasonably accurate enough extent, though we have seen that this was the approach taken in (Bender et al., 2012). As all the different methods (bar SoD) have a complexity of $O(nm^2)$ where n is the size of the data, and $n < m$, the authors notably point out that no gross approximations should be made as more competent methods are computationally equivalent, and as such point towards their notes on future work to outperform the existing state of the art. As such, we would wish to explore combining the **Partially Independent Training Conditional approximation** with "the most powerful selection method for the inducing inputs."

CHAPTER 3

Machine Learning Background

Things to (possibly/likely) cover in this background chapter

(1) normal distribution
(2) variance
(3) bias, error
(4) regression
(5) prior
(6) posterior
(7) marginal likelihood
(8) log likelihood

CHAPTER 4

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.

4.1 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:

$$f(x) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{4.1}$$

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:

$$cov(f(\mathbf{x_p}), f(\mathbf{x_q})) = k(\mathbf{x_p}, \mathbf{x_q}) = \exp(-\frac{1}{2}|\mathbf{x_p} - \mathbf{x_q}|^2) \tag{4.2}$$

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 **??** 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:

$$k(\mathbf{x_p}, \mathbf{x_q}) = \sigma_f^2 \exp(-\frac{1}{2l^2}(\mathbf{x_p} - \mathbf{x_q})^2) + \sigma_n^2 I \tag{4.3}$$

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 **??** 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):

$$\mathbf{f}_*|\mathbf{x}_*, X, \mathbf{f} \sim \mathcal{N}(\mathbf{k}_* K^{-1}\mathbf{f}, k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_* K^{-1} K_*) \tag{4.4}$$

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:

$$\bar{f}_* = \mathbf{k}_*{}^T (K + \sigma_n^2)^{-1}\mathbf{y} \tag{4.5}$$

$$\mathbb{V}[f_*] = K(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*{}^T (K + \sigma_n^2)^{-1}\mathbf{k}_* \tag{4.6}$$

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 $\mathbf{f}$:

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f}, X)p(\mathbf{y}|X)d\mathbf{f} \tag{4.7}$$

Taking the log of this then gives:

$$\log p(\mathbf{f}|X) = -\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{4.8}$$

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:

$$\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|X,\theta) = \frac{1}{2}\text{tr}((\alpha\alpha^T - K^{-1})\frac{\partial K}{\partial\theta_j}) \text{ where } \alpha = K^{-1}\mathbf{y} \tag{4.9}$$

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.

4.2 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

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

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

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

4.3 Gaussian Process Classification

To perform Gaussian Process Classification, multiple Gaussian Process Regressors are used for each possible label using a one-vs-all approach, where every label at each data point is assigned both a probability and variance.

(expand)

4.4 Subsampling for Gaussian Process experiments

(may need to take this section out - GPy wrapper deals with thousands of points no problem, no longer need to subsample with only 4000 after downsampling the data)

Due to the $O(n^3)$ complexity of training a Gaussian Process Classifier, using all 16502 points was infeasible, so it was necessary to use only a subsample of the training data. As can be seen in the above histograms (reference the figure instead. may need to combine them into one), the distribution of classes in both the simplified and non-simplified versions was very uneven. As a result of this skew, randomly sampling the the training data to fit our GP classifier against resulted in worse results than samplying an equal *number* of points for each class. To obtain a reasonably well-performing set of 1000 points (the number chosen to obtain a balance between performance and time required), 10-fold cross validation was performed on random subsets of this size. To obtain the 1000 datapoints, both stratified sampling, as well as obtaining ratios of labels matching those in the training set were used. After 200 runs, the set with the best performance was used for the remaining GP experiments.

4.5 Gaussian Process Approximation

However, although significant limitations exist in terms of the number of training datapoints used when using a single Gaussian Process Classifier, there exist 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 is *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.

4.5.1 Product of Experts and Variations

(EXPAND explanations in this section)

Product of GP Experts

$$\mu_*^{poe} = (\sigma_*^{poe})^2 \sum_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{4.10}$$

$$(\sigma_*^{poe})^{-2} = \sum_k \sigma_k^{-2}(\mathbf{x}_*) \tag{4.11}$$

Generalised Product of GP Experts

$$\mu_*^{gpoe} = (\sigma_*^{gpoe})^2 \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{4.12}$$

$$(\sigma_*^{gpoe})^{-2} = \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*) \tag{4.13}$$

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.

4.5.2 Bayesian Committee Machines and Variations

Bayesian Committee Machine

$$\mu_*^{bcm} = (\sigma_*^{bcm})^2 \sum_{k=1}^{M} \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{4.14}$$

$$(\sigma_*^{bcm})^{-2} = \sum_{k=1}^{M} \sigma_k^{-2}(\mathbf{x}_*) + (1-M)\sigma_{**}^{-2} \tag{4.15}$$

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

Robust Bayesian Committee Machine

$$\mu_*^{rbcm} = (\sigma_*^{rbcm})^2 \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*)\mu_k(\mathbf{x}_*) \tag{4.16}$$

$$(\sigma_*^{rbcm})^{-2} = \sum_k \beta_k \sigma_k^{-2}(\mathbf{x}_*) + (1 - \sum_{k=1}^{M} \beta_k)\sigma_{**}^{-2} \tag{4.17}$$

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

(Deisenroth, 2015)

CHAPTER 5

Multi-label Habitat Mapping

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

5.1 Multinomial Distribution

equations, description

5.2 Dirichlet Distribution

descriptions

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

$p(\theta) = \frac{1}{\beta(\alpha)} \Pi_{i=1}^{n} \theta_i^{\alpha_i - 1} I(\theta \in S)$ 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

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

5.3 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:

$$\sum_{n=1}^{N}[\log(M_k) - \sum_k \log(c_k!) + \log\Gamma(\sum_k \alpha_k(x_n)) - \log\Gamma(\sum_k c_{nk} + \alpha_k(x_n))]$$
$$+ \sum_{n=1}^{N}\sum_{k=1}^{K}[\log\Gamma(c_k + \alpha(x)) - \log\Gamma(\alpha_k(x_n))]$$
$$+ \sum_{k=1}^{K}[-\frac{\phi}{2}\log(2\pi\phi) - \frac{1}{2}w_k^T\phi\mathbb{I}w_k] \quad (5.1)$$

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

$$\partial \frac{\log p(c,x)}{\partial w_k} = \sum_{n=1}^{N} x_n \alpha_k(x_n)[\psi(\sum_l \alpha_l(x_n)) - \psi(\sum_k c_{nk} + \alpha_k(x_n))] + \sum_{n=1}^{N} x_n \alpha_k(x_n)[\psi(c_{nk} + \alpha_k(x_n)) - \psi(\alpha_k(x_n))] - \frac{1}{\phi} w_k \quad (5.2)$$

explain all the symbols here

5.3.1 Using MCMC instead of MAP

(explain why)

5.4 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.

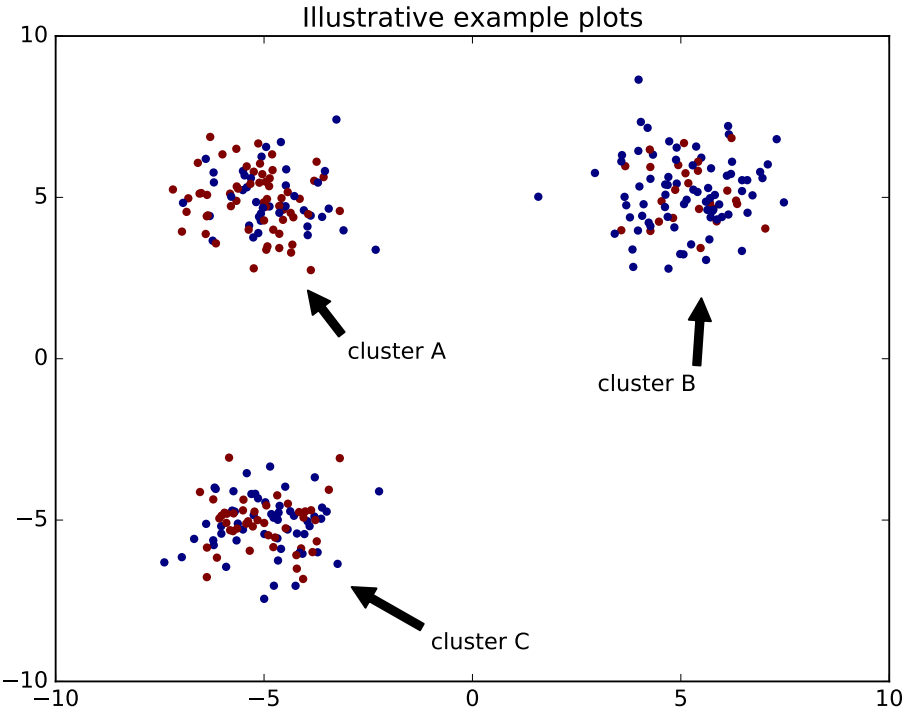


FIGURE 5.1: Plots of the three clusters, with labels taking on the argmax of each point

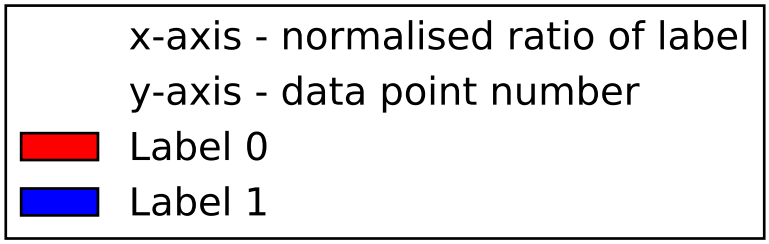


FIGURE 5.2: Legend/axes for the following histogram plots showing distribution of labels at each point

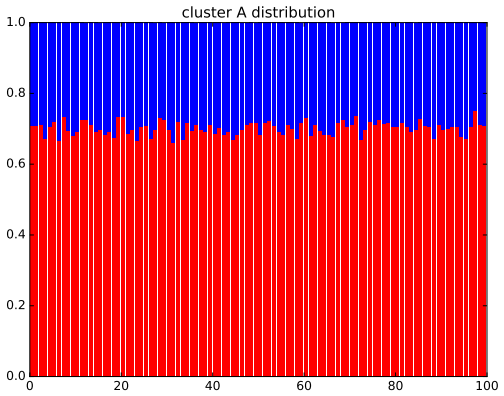


FIGURE 5.3: Label distribution of cluster A

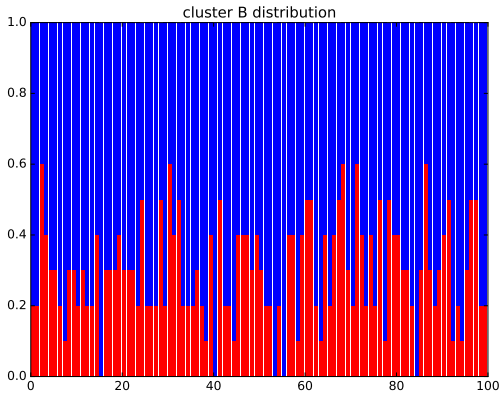


FIGURE 5.4: Label distribution of cluster B

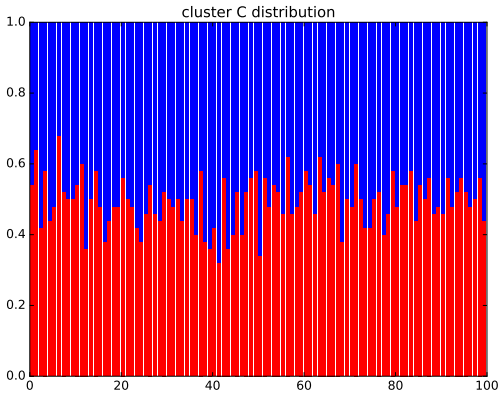


FIGURE 5.5: Label distribution of cluster C

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.

5.4.1 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)

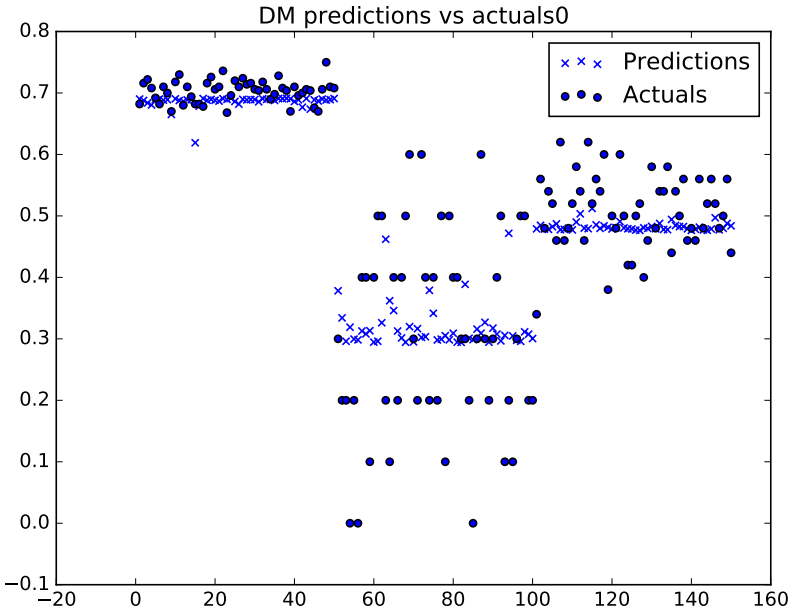


FIGURE 5.6: DM Label distribution of label 0

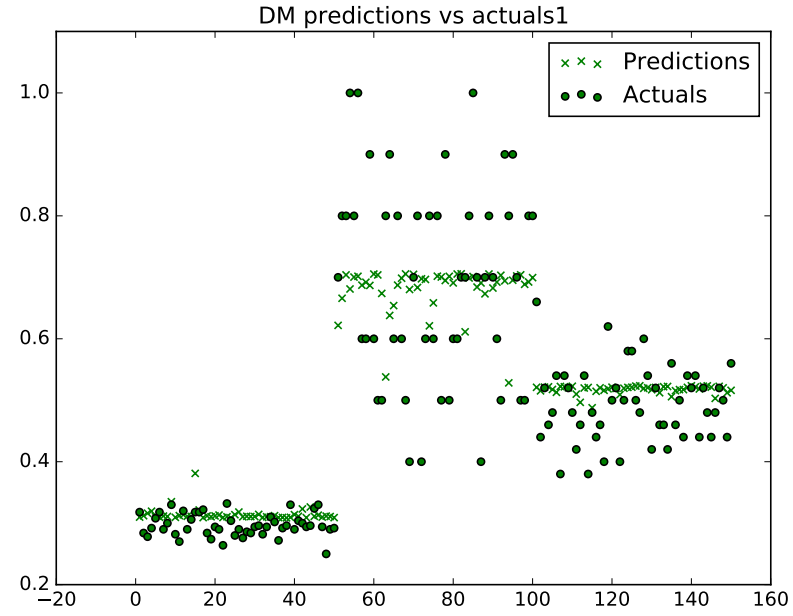


FIGURE 5.7: DM Label distribution of label 1

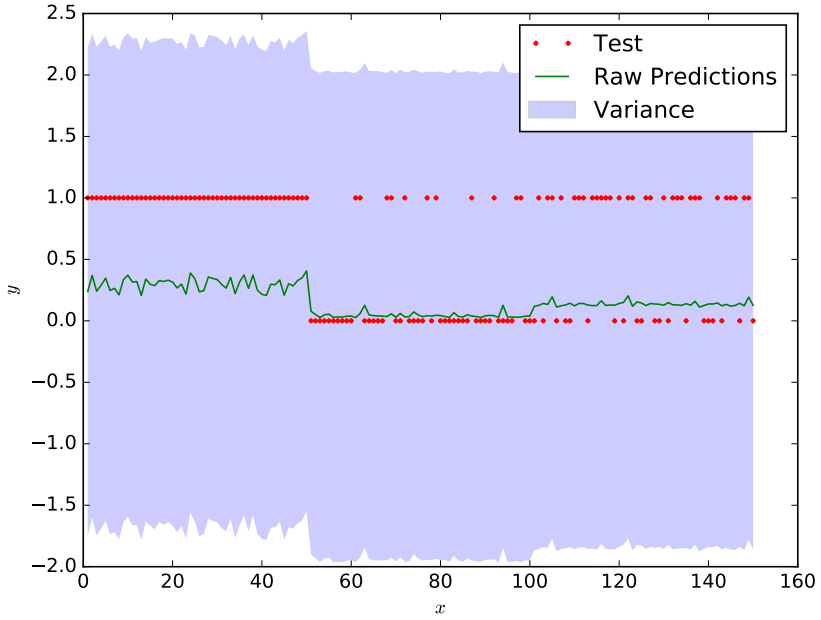


FIGURE 5.8: OvR GP performance with variance for label 0

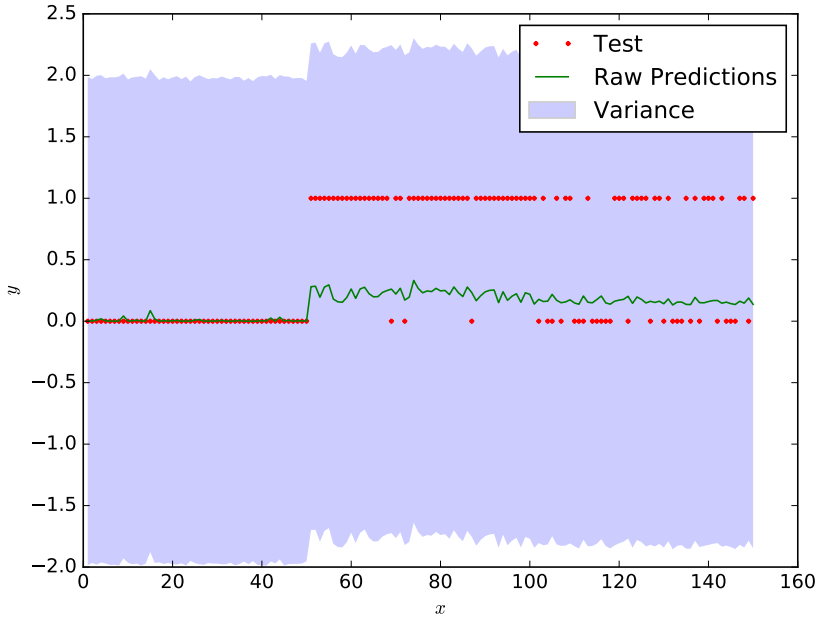


FIGURE 5.9: OvR GP performance with variance for label 1

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 6

Experiments and Results

To identify whether the Dirichlet Multinomial Regression method proposed can provide richer and more valuable information than single-output or deterministic methods can alone, we ran experiments on the data obtained from the ACFR's Sirius AUV and Schmidt's Falkor. The main machine learning algorithms' performance which we tested were Gaussian Process Classification, and Dirichlet Multinomial Regression. In this chapter, we detail the experiments designed to display the benefits of a Gaussian Process Classifier's probabilistic output, as well as the label distributions of a Dirichlet Multinomial Regressor and their respective results. (flimsy, fix)

6.1 Training Data

To perform our experiments, bathymetry data and images of Scott Reef Central were used (the reef in the centre of Figure 6.1). The bathymetry data was collected using Eric Schmidt's Falkor (at the locations in Figure 6.2), 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). On close inspection, the UTM coordinates in the training set do not correspond to the original data available from (squ, 2016) - 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.)

6.2 Data Preprocessing

6.2.1 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 siginficantly higher resolution than the 2009 dataset (calculate how much perhaps). Two methods of downsampling in particular were tested. The first coarser approach involved simply taking the space in which the data was collected and placing grids of fixed size over them as in **??**, binning all points falling within each grid into a single datapoint. Each of these data points contained multiple points from the original dataset with their own counts for each of the possible labels, so the downsampled points simply took the sum of all the label counts in each fixed grid.

The second summed label counts in the same way, but clusters were instead formed by first calculating the full dendrogram on the 16502 entries in the training data, and forming groups such that none had more than 5 of the original points within them, and the sub-clusters (at each level of the dendrogram) were no more than a 21 metres away from one another. As can be seen in **??**, the gradual merging into the single supercluster was quite consistent, indicating the original datapoints were mostly evenly distributed.

For a fair comparison between Gaussian process classification and dirichlet multinomial regression, the downsampled data was used to train the GPs as well - although this seems like an unnecessary handicap to the GP, it is more appropriate considering that one of the aims here

FIGURE 6.1: Aerial shot of Scott Reef from (National Aeronautics and Space Administration(NASA), 1996)

is to demonstrate what sort of information can be gained from a DM vs. a GP, given the same *raw* data.

(cut down this section (probably down to one paragraph) - and also talk about how for the single label case, instead of summing counts for the new 'label', labels were chosen at random (for more variance in the GP which is otherwise too certain about everything being sand))

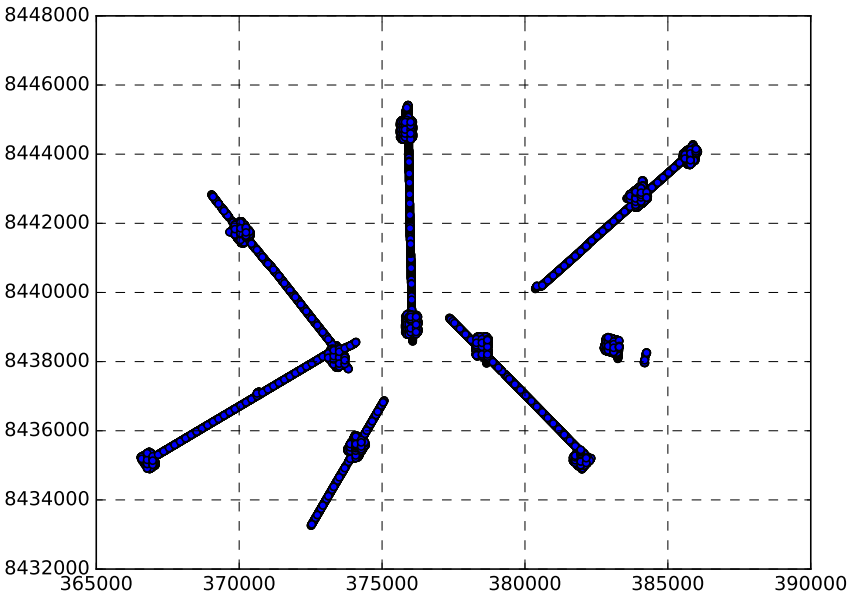


FIGURE 6.2: Fixed-sized grids placed over training data

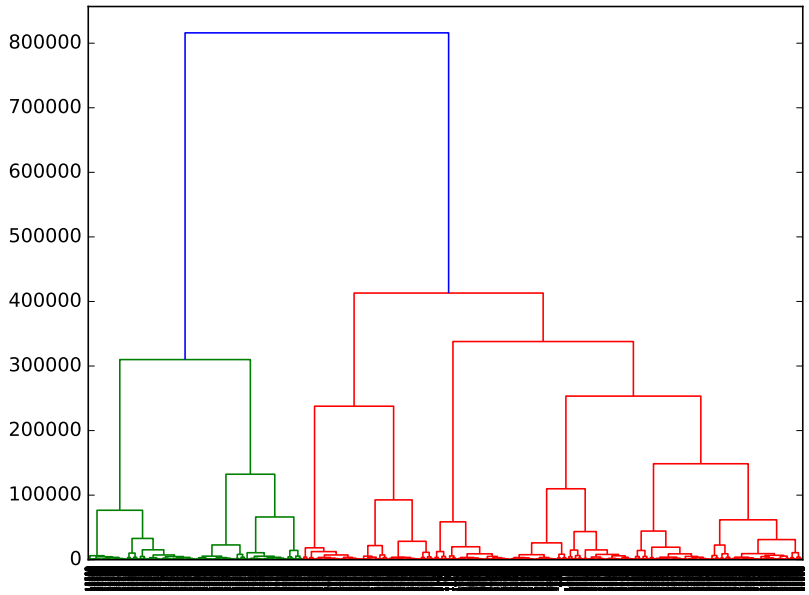


FIGURE 6.3: Dendrogram of training data

6.2.2 Simplifying labels

Another step that was considered during experiments 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 6.7 and Figure 6.8), with the occurrence of some too insignificant for any machine learning algorithms to pick up, they were simplified in collaboration with ecological experts, who manually identified which of the 24 labels were in fact of the same class - for example, 5 separate classes of coral may have been indistinguishable to the average person, and were hence grouped into a single label. This allowed the near-non-occurring labels to be grouped together with more commonly occurring ones, whilst also allowing a different level of granularity in training models/forming predictions that could be used if only an approximation equivalent to observable human differences of an area's benthic map were required. 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 6.1: Full-simplified label mappings label mappings - sand, coral, patchy coral, (?) halameda, rhodliths

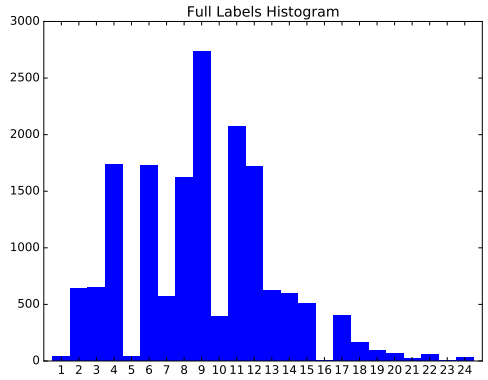


FIGURE 6.5: Distribution of labels in original dataset

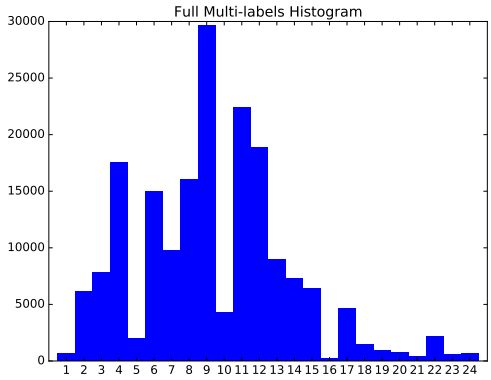


FIGURE 6.6: Distribution of labels in multi-label outputs

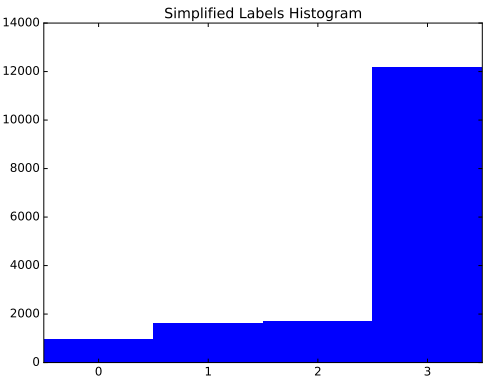


FIGURE 6.7: Distribution of simplified labels in original dataset

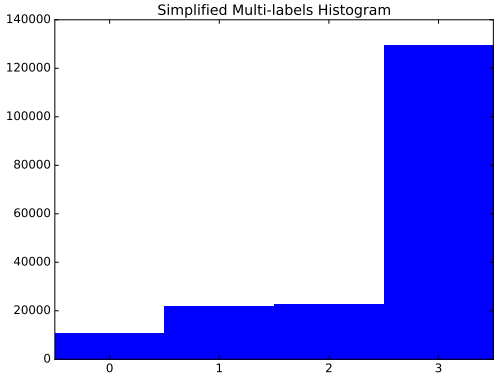


FIGURE 6.8: Distribution of simplified labels in multi-label outputs

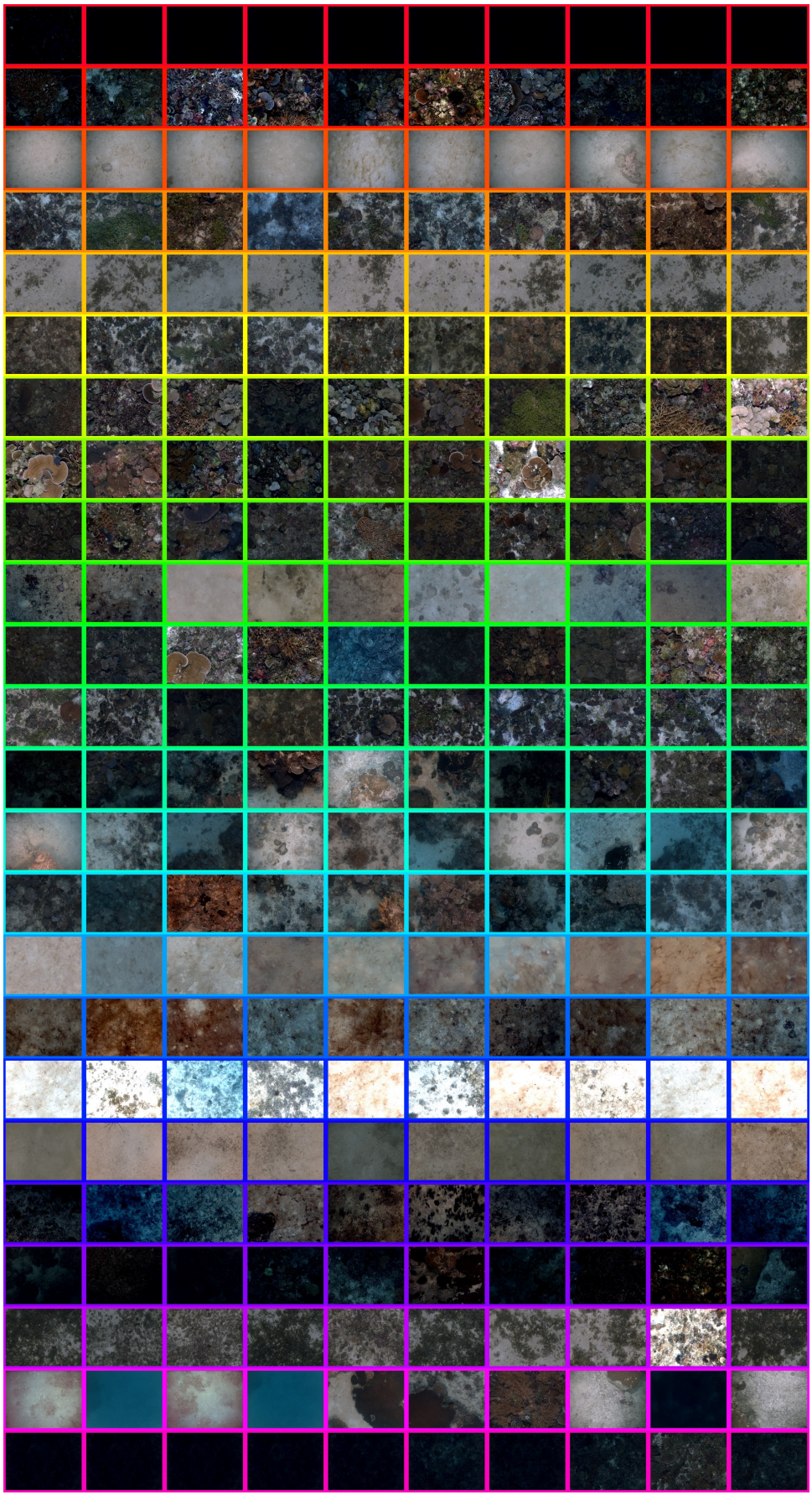


FIGURE 6.4: Samples of images from each of the full 24 classes

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.

6.2.3 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, there was reason not to include the coordinates themselves in the feature space. Whilst it does make sense that in a natural environment, areas that were spatially near to one another 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 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 having other bathymetry information such as depth, aspect, etc.

(argument/s here alone is/are weak. for simplified labels using coords is still much better by a similar (small) margin, do some reading to back this up properly)

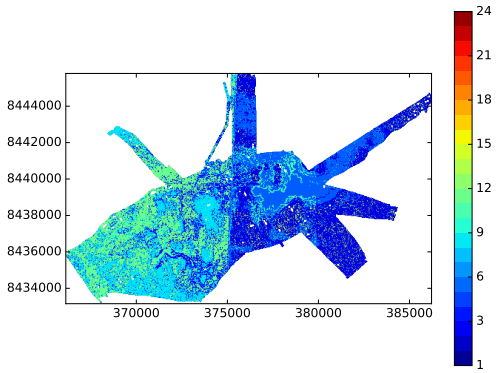


FIGURE 6.9: Full predictive map using Random Forests including coordinates as features

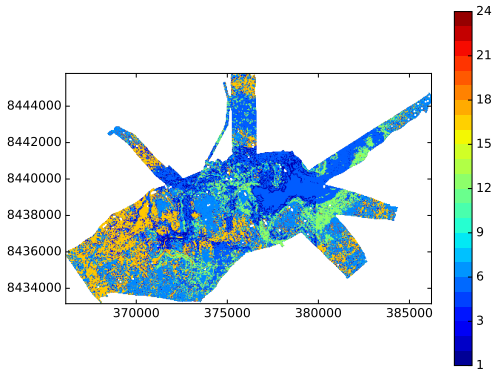


FIGURE 6.10: Full predictive map using Random Forests excluding coordinates as features

6.2.4 Preprocessing and Feature Projection

To maximise performance (in terms of minimising error/maximising the correctness metrics used) of the algorithms used across the experiments, a number of preprocessing steps were taken to ultimately improve the performance of 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 further minimise bias and variance(maybe add to this technically), the original feature set (which from this point onwards refers to all the non-coordinate features) was projected into degree-2 space with a 1-bias term. For example, given features x_0, x_1, x_2, x_3 is projected to $1, x_0, x_1, x_2, x_3, x_0^2, x_1^2, x_2,^2, x_3^2$. This projection was chosen from comparing the performance of different projections

(ref plots)

(show plots)

6.3 Results

As evident from above, adding the square (^{2}) of each feature with a 1 regulariser (figure out what 1 column should be called) yielded in performance that was negligibly lower ($\sim 1e-5$) than projection into full quadratic space, whilst requiring considerably fewer features - 19 for the former, vs. 55 for the latter. While this difference isn't significant for GPs (perhaps provide a time benchmark for this too), it has a significant effect on running time later on when using Markov Chain Monte Carlo (MCMC) to obtain draws of the weights of the Dirichlet Multinomial Regressor - when working with the simplified 4-label case, there are $19 * 4 = 76$ vs. $55 * 4 = 220$ weights, i.e. dimensions to deal with, whereas the full 24-label case would involve $19 * 24 = 456$ and $55 * 24 = 1320$ weights respectively, which would result in a very significant increase in computing power needed for the MCMC chains to converge in the latter case, relatively speaking.

The results from the Experiments detailed in Chapter 4 are listed below. The range of possible class values in some cases have been stretched beyond the existing class labels so that values align between different outputs to allow for easy, direct visual comparison. Note that the results to the above experiments will include those of both non-downsampled and downsampled results, as well as the full set of 24 labels as well as simplified ones.

Due to the low occurrence of some labels in the original dataset though, they have ended up being ommitted in predictions - these are excluded from the colour schemes of the benthic maps generated, so that those that do occur can be given more distinct colours from one another as to better differentiate between the habitats of a map, as well as allow a consistent comparison of across different maps.

In this section, the performance of common machine learning algorithms, namely kNN, Logistic Regression, Random Forest, and SVM are explored first, to provide a comparison to the later, more complex algorithms.

Algorithm	10F-CV F1	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 6.2: Performance of common machine learning models

6.4 Deterministic Methods

To first set a baseline for predictions over the bathymetry dataset, we look at the performance of some deterministic methods with respect to their weighted (by number of instances per label) f-scores. 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.

While the accuracy of the Logistic Regressor, kNN, and Random Forest Classifier are reasonable (above 0.75), the former's f1-score is very poor at 0.33, and the latter two sit just below 0.5, which is an equally undesirable result. Looking at the ratio of available labels in the down-sampled 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 weighted f1 score of 0.6503 - highlighting the fact that these simpler models are not able to produce results that confidently outperform simply guessing one label for any given datapoint. (ref figure) below shows the predictions of each of the above predictions on the full query data for the 4 and 24-label data respectively.

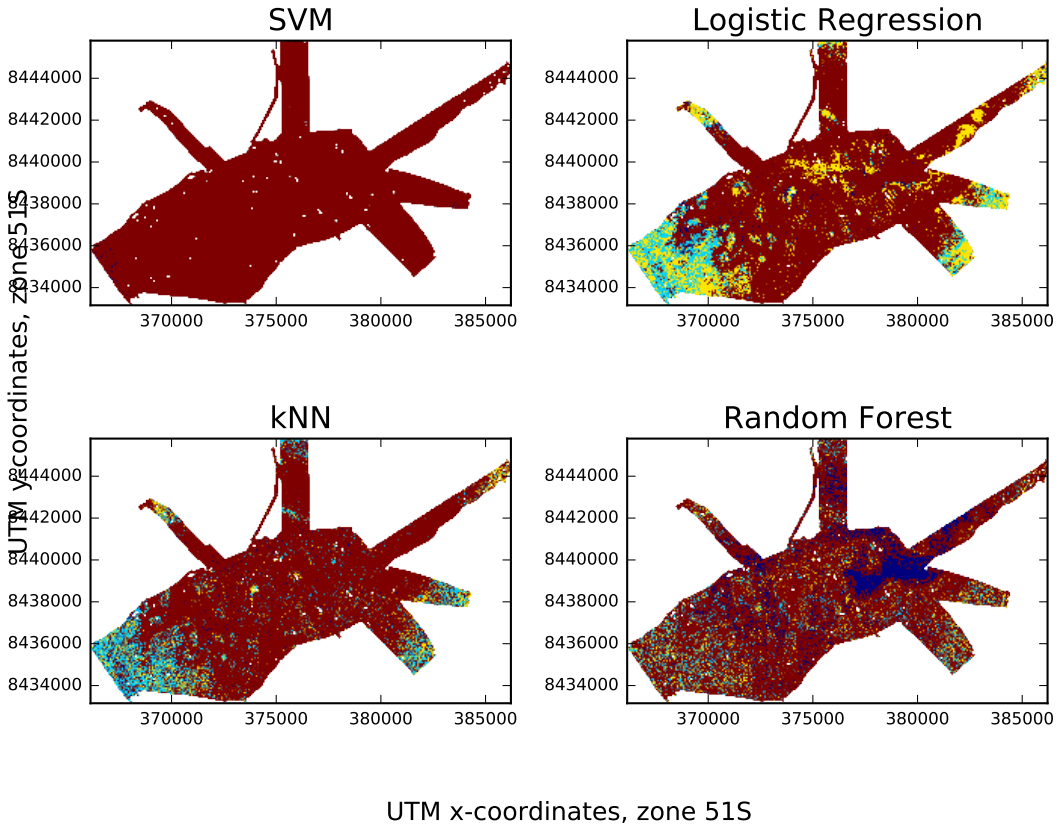


FIGURE 6.11: Full predictive map using Random Forests excluding coordinates as features

6.5 Gaussian Process Classification

transfer all the results from markdown

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

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

(maps of 4-label full predictions)

(maps of all-label full predictions)

6.6 Dirichlet Multinomial Regression

6.6.1 Parameter Selection

To select an optimal set of parameters for the dirichlet multinomial, Markov Chain Monte Carlo (MCMC) was used to draw samples from the posterior distribution (refer to equation?) over $3,000,000$ runs, with the maximum a posteriori estimate used as the starting value for the weights. To select the single best set of weights from the sequence of chains, every single

one was evaluated by being used to do Dirichlet Multinomial regression, where the weights that resulted in the lowest predictive variance (average over all variances) was considered to be the best set of parameters. The weights that corresponded to the lowest average variance also corresponded to the lowest average error compared to the original normalised weights. After the $3,000,000$ runs, the MCMC in both cases (the simplified 24 labels, as well as the full set) was considered to have converged, as the Gelman and Rubin ($\hat{r}$) convergence statistics were calculated to be (? and ?), both very close to the ideal value of 1.0, Furthermore, for the 24-label case

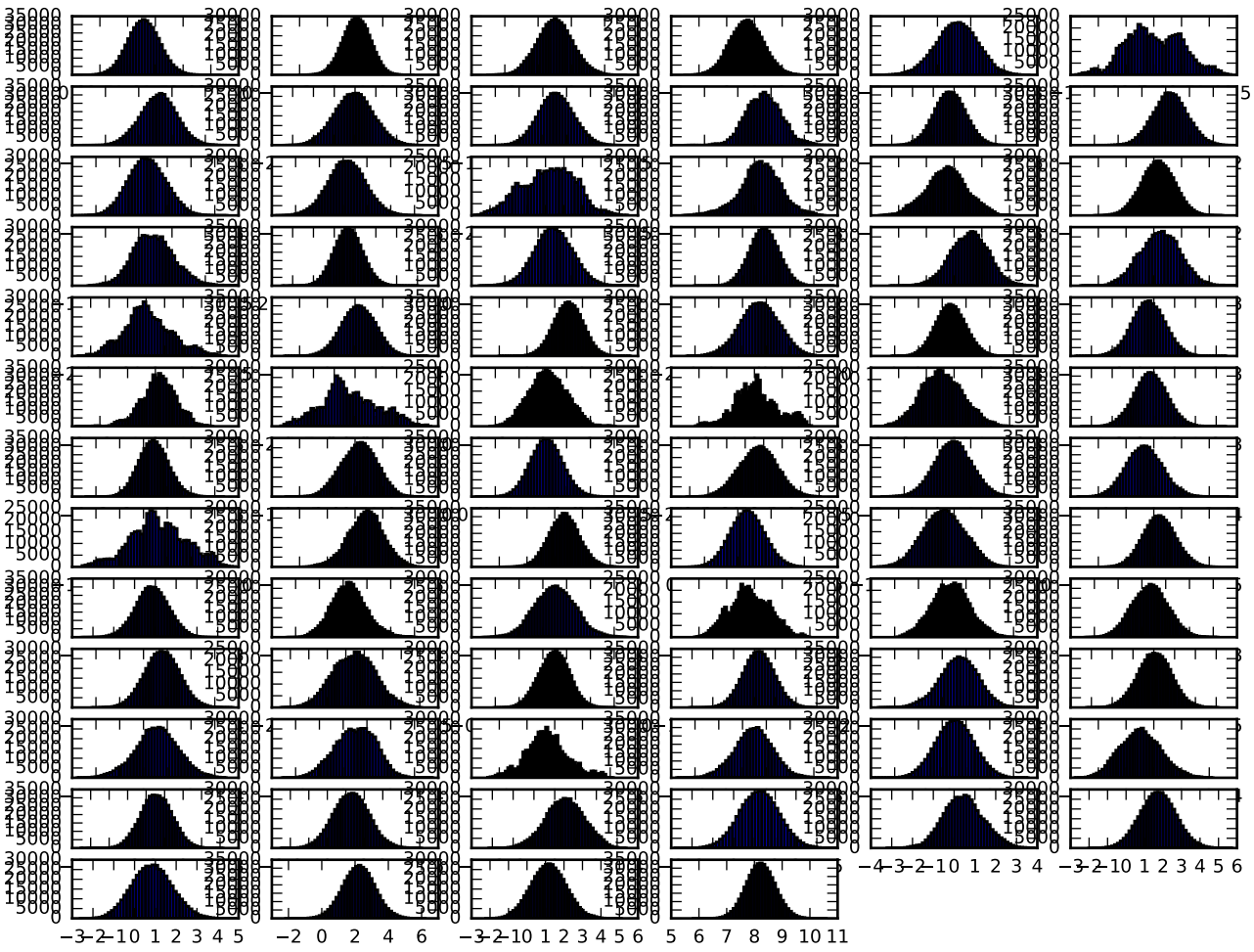


FIGURE 6.12: MCMC weights for 4-label, 19-dimension data case (need to separate this into separate images, possibly remove axis ticks)

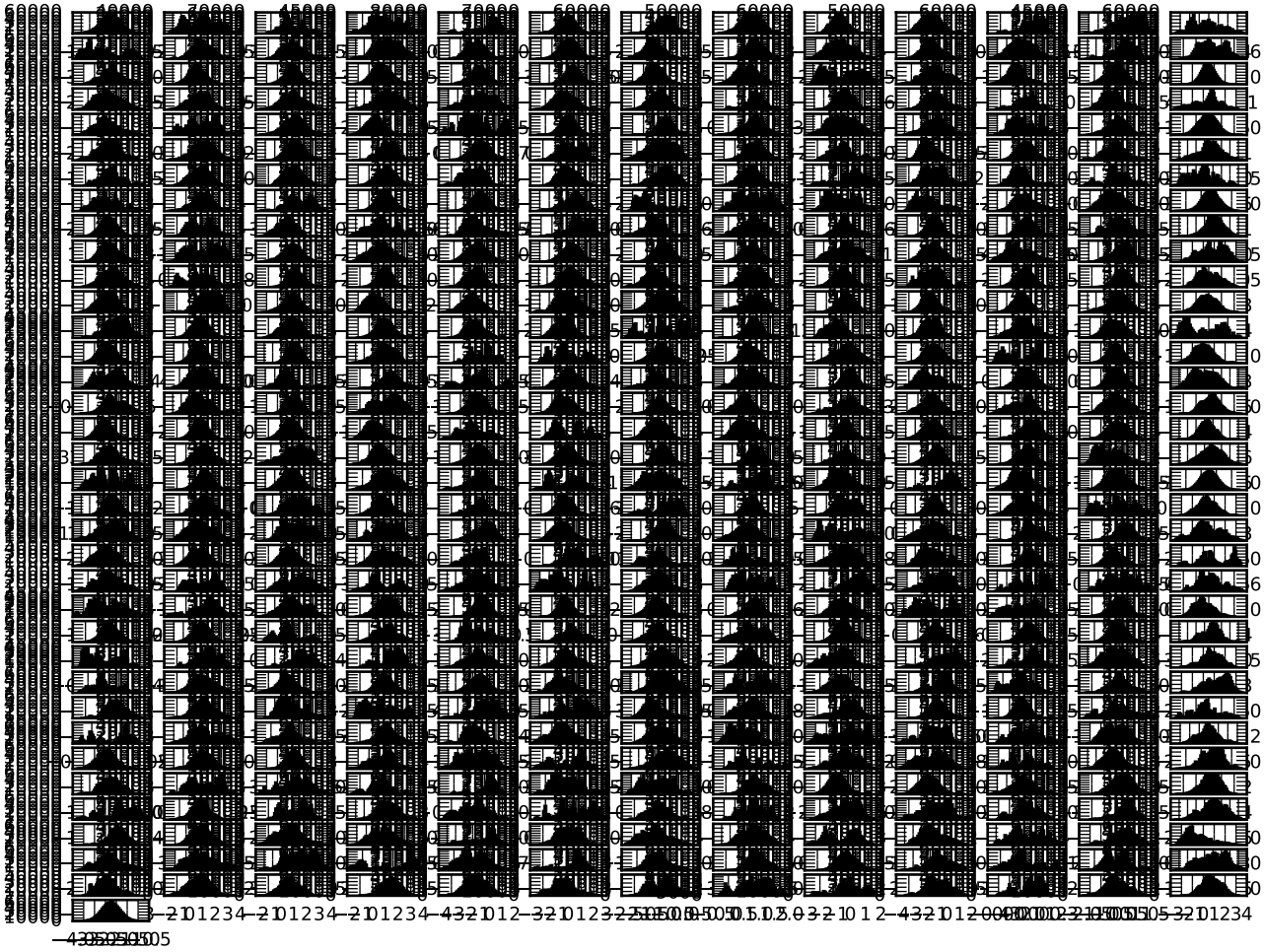


FIGURE 6.13: MCMC weights for 4-label, 19-dimension data case (need to separate this into separate images, possibly remove axis ticks)

Features	Root Mean Squared Error
Original features, using coordinates	0.3360700584111156
Original features, not using coordinates	0.33676707345298545
Quadratic projection, using coordinates	0.19695170465540882
Quadratic projection, not using coordinates	0.19726479764111976

TABLE 6.3: Dirichlet Multinomial Regression results

In line with what was observed in the DM illustrative example in Table 5.1, performance was better when taking the quadratic projection of the original features - though in this case, the improvement was much more significant, namely by 41.4% compared to the 6.5% in the illustrative example. (numbers here will change)

As a means of effectively visuailsing the separate labels, we need to look at the normalised distribution of habitat classes for each label separately. This allows initial observations to be made of where certain labels are more abundant than others. Immediately, we can see the concrete advantages of using a multi-output model, as individual points no longer contain only a single label - Figure 6.14 for example, shows that the majority of the reef is predominantly sand (label 3) with smaller proportions of the other labels, with the bottom-left region being mostly an even combination of labels 1, 2 (get actual labels) and scattered sections of sand. Using the approaches previously explored, it would not have been directly inferable (without extra processing of labels/results) that the sand-dominant areas also contained small amounts of other labels.

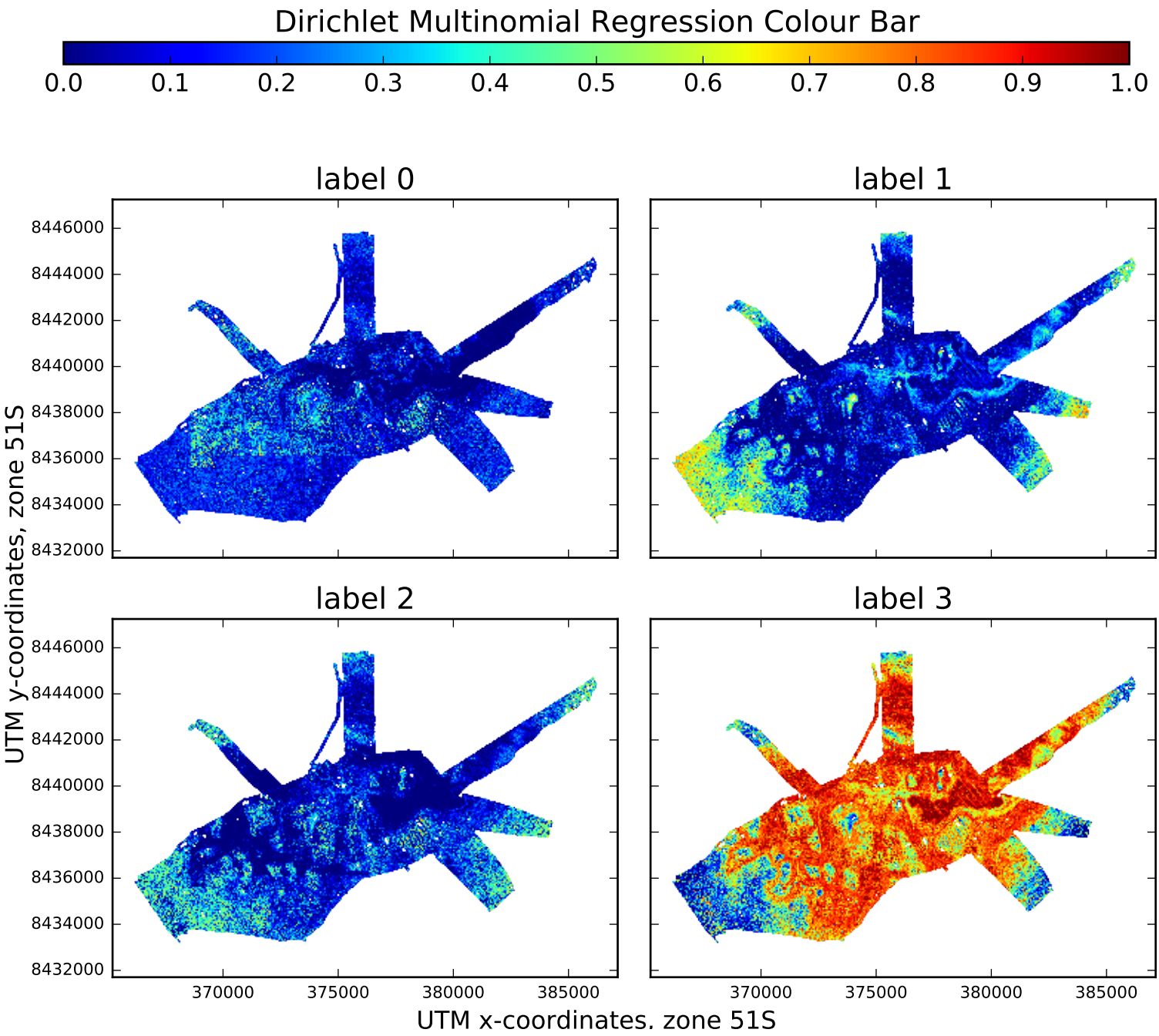


FIGURE 6.14: Distribution heatmaps over each label (in the simple 4-label case) for Dirichlet Multinomial Regressor output on query points

In contrast to the GP above where the uncertainty was greatest when there were even distributions of labels, 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'.

summarise and plot the variances here

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.

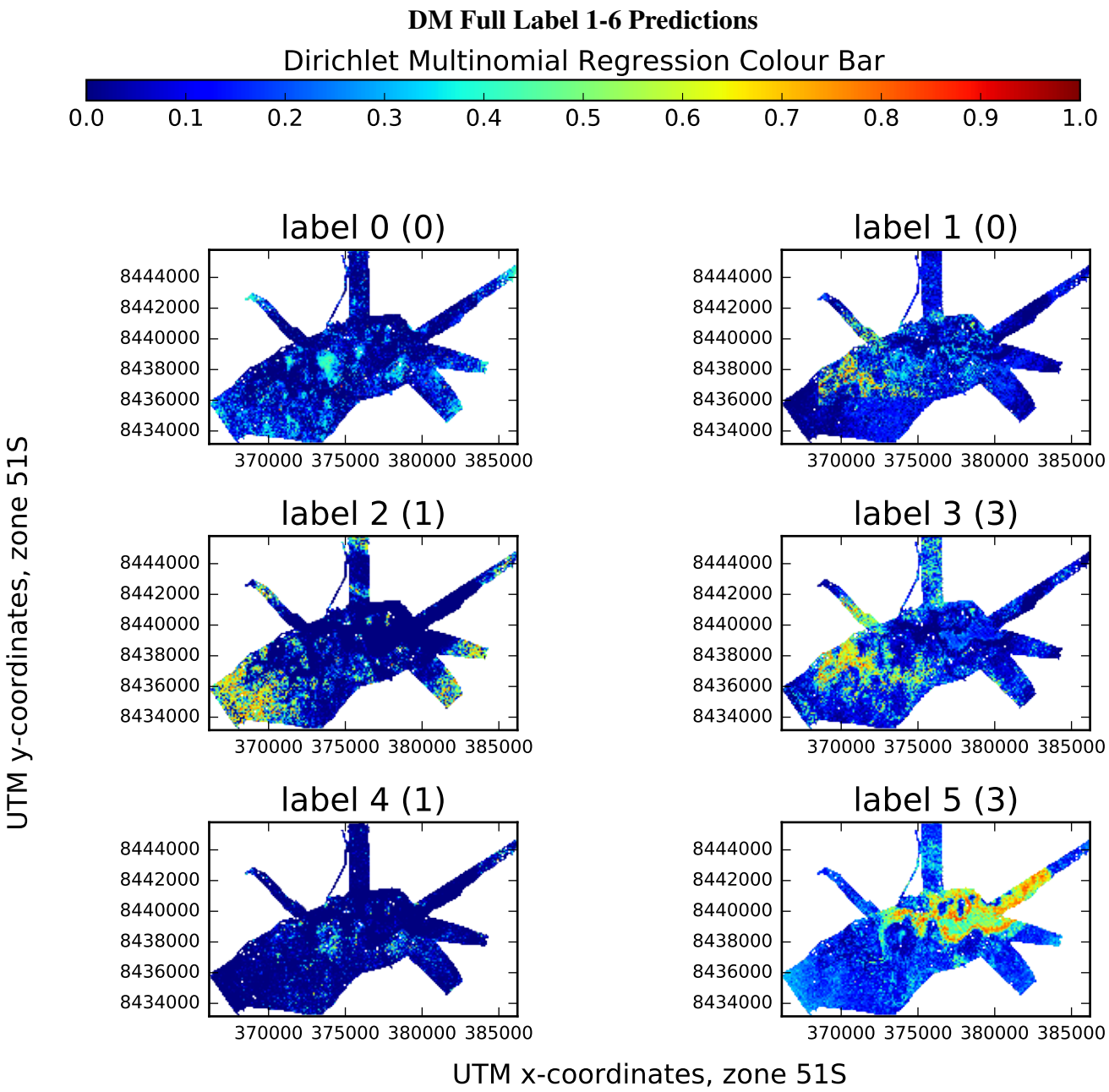


FIGURE 6.15: Distribution heatmaps over labels 1-6 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

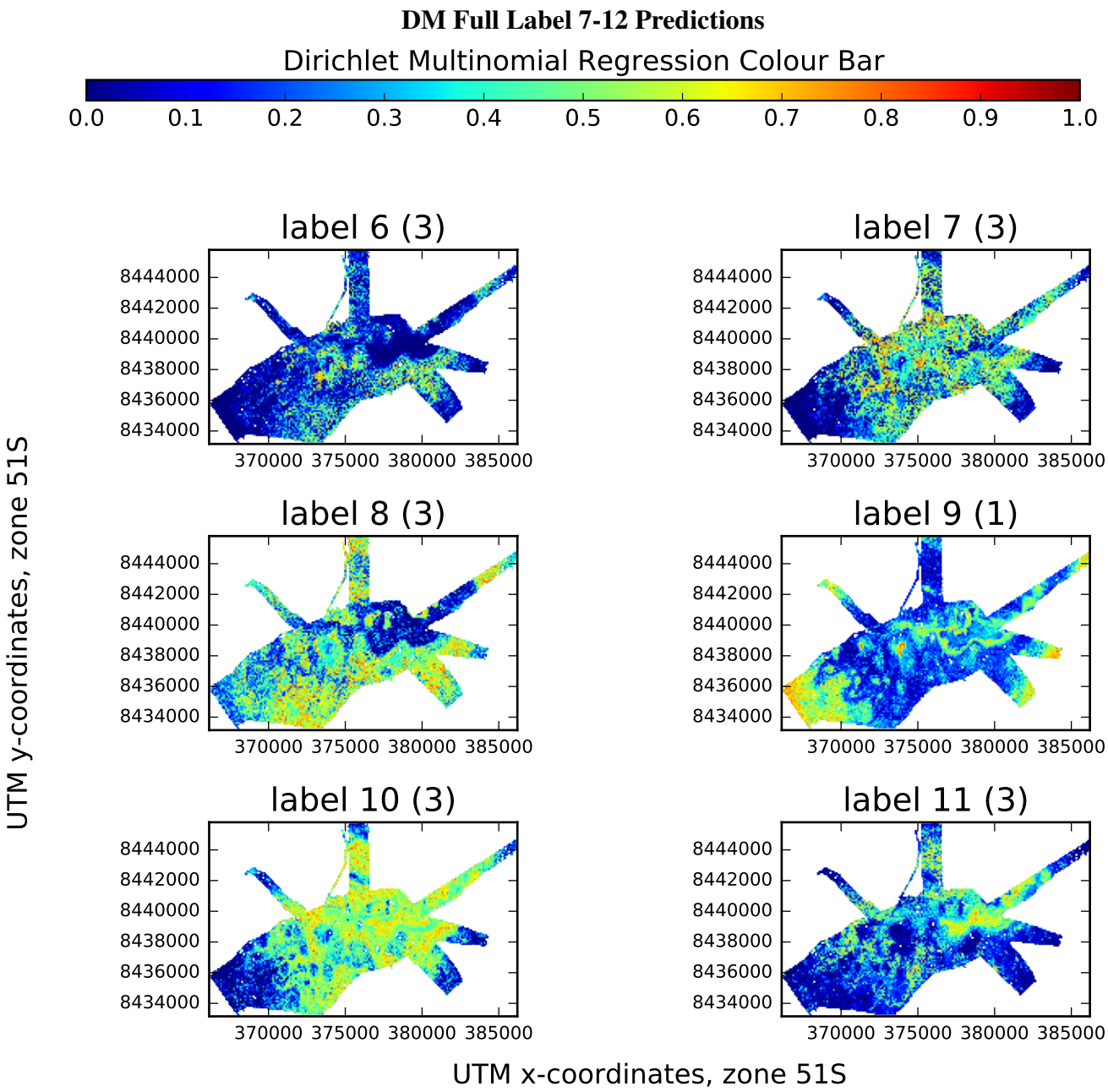


FIGURE 6.16: Distribution heatmaps over labels 7-12 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

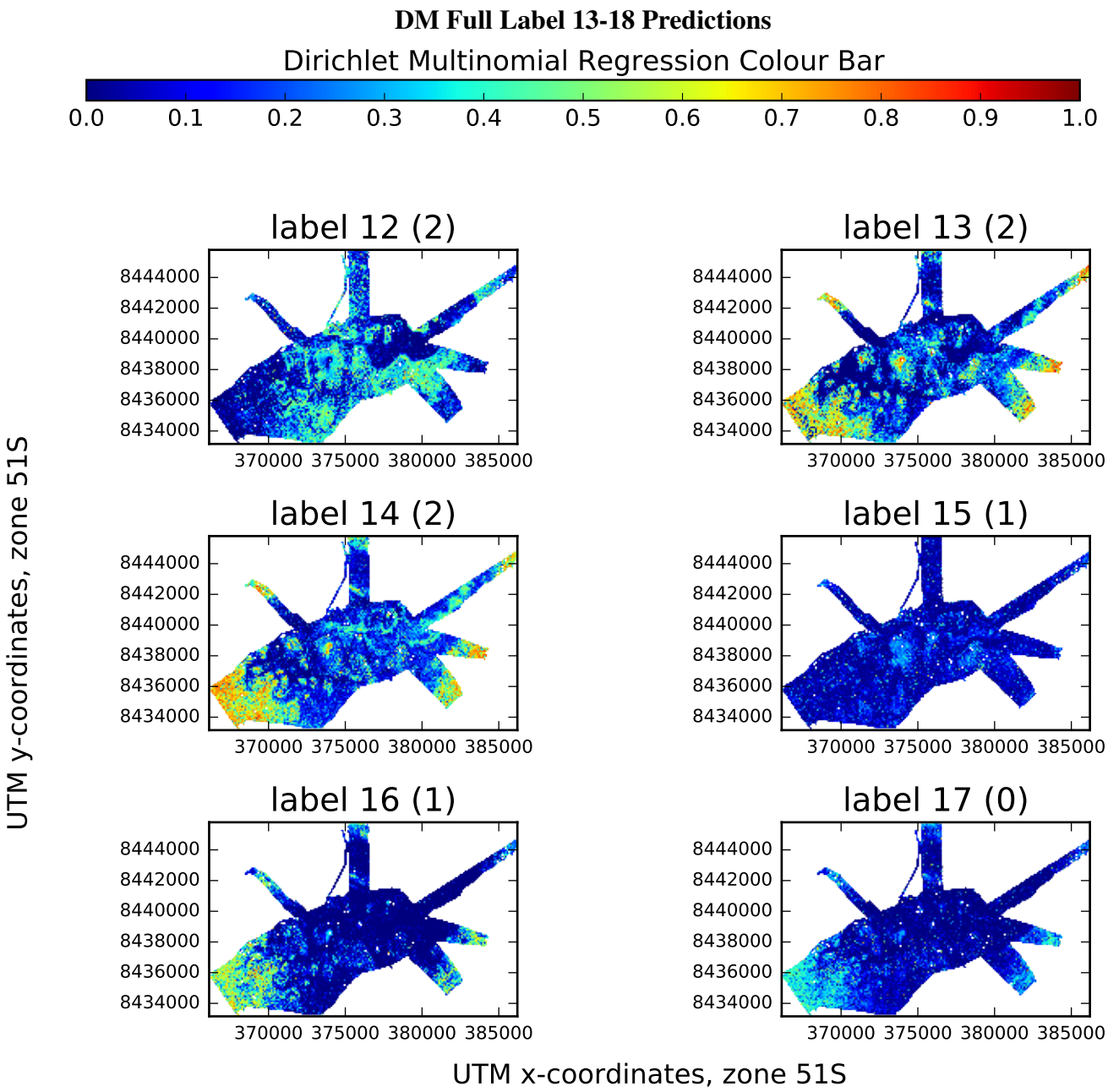


FIGURE 6.17: Distribution heatmaps over labels 13-18 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

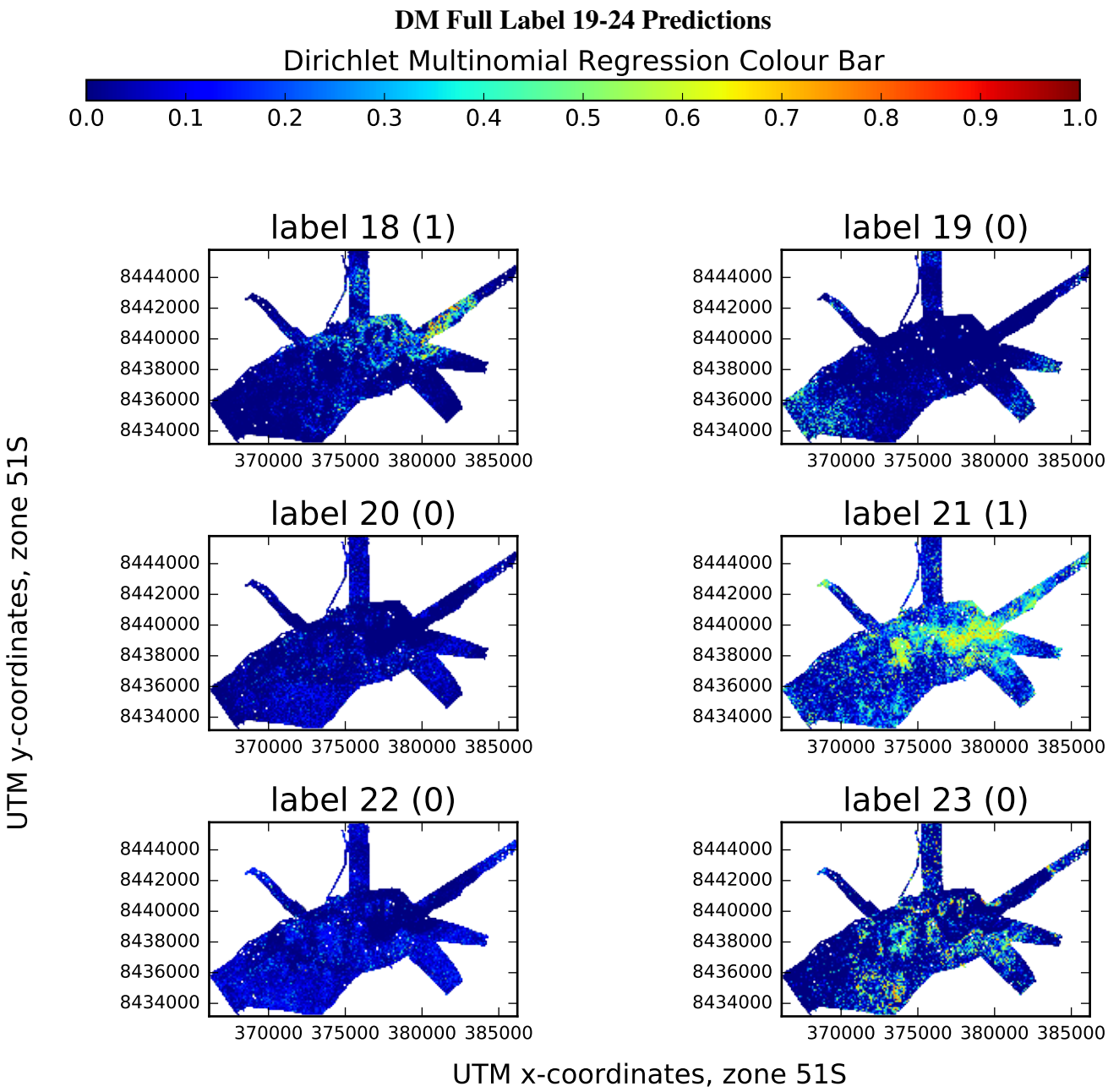


FIGURE 6.18: Distribution heatmaps over labels 19-24 (in the full 24-label case) for Dirichlet Multinomial Regressor output on query points

6.6.2 Biodiversity

highlight areas with biodiversity, particular co-occurring species, etc.

Another beneficial aspect of DMs are that they can inherently provide 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, assigning each a particular portion, with a margin of error. For example, we could easily locate areas with mixes of (change here for a good example) labels 5, 9, and 14, with each having an even split with a margin of 0.05 - this would result in us searching the predictions with the conditions where label 5 has a predictive distribution $[0.28, 0.38]$, as would labels, 9, 14.

(some plots of these - pick ones that have 'nice' graphs)

The above scenario assumes, however, that an ecologist (or anyone using the data) is already aware of the proportions of habitats they are searching for, which may not be the case if (describe a scenario here where the application is to detect changes in biodiversity over time)

CHAPTER 7

Evaluation and Discussion

---

old note - these will likely be in results/described in experiments Looking at distributions of the GPs, dirichlet multinomial draws and how they perform beyond just taking argmax, etc. should go here

7.0.1 Limitations

- 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)

CHAPTER 8

Conclusion

The conclusion goes here.

8.1 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

Bibliography

2016. Squidle projects. `http://squidle.acfr.usyd.edu.au/viewproject#map`.

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.

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

Peter J. Bickel and Elizaveta Levina. 2008. Regularized estimation of large covariance matrices. *The Annals of Statistics*, pages 199–227.

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.

Robert J. Diaz, Martin Solan, and Raymond M. Valente. 2004. A review of approaches for classifying benthic habitats and evaluating habitat quality. *Journal of Environmental Management*, 73:161–181.

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.

Arman Melkumyan and Fabio Ramos. 2009. A sparse covariance function for exact gaussian process inference in large datasets. *International Joint Conference on Artificial Intelligence*, 9.

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.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

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

Andrew Y. Ng and Michael I. Jordan. 2002. On discriminative vs.generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841.

Food & Agriculture Organisation of the United Nations. 2004. *The State of World Fisheries and Aquaculture*.

Joaquin Quinonero-Candela and Carl Edward Rasmussen. 2005. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959.

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

A. Rattray, D. Ierodiaconou, L. J. B. Laurenson J. Monk, and P. Kennedy. 2014. Quantification of spatial and thematic uncertainty in the application of underwater video for benthic habitat mapping. *Marine Geodesy*, 37:315–336.

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.

Paul V R Snelgrove. 1994. Animal-sediment relationships revisited: Cause versus effect. *Oceanography and marine biology*.

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.