UNIVERSITY OF SYDNEY

Informative Seafloor Exploration for Benthic Habitat Mapping

by

Kelvin Y.S. Hsu

A thesis submitted in partial fulfillment for the degree of
Bachelor of Engineering (Mechatronic - Space) &
Bachelor of Science (Advanced Mathematics)

in the

Faculty of Engineering & Information Technologies
School of Aerospace, Mechanical, and Mechatronic Engineering

September 2015

Declaration of Authorship

I, Kelvin Y.S. Hsu, declare that this thesis titled, 'Informative Path Planning with Gaussian Process Classifiers for Seafloor Exploration' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

“三人行，必有我師焉”

孔子

UNIVERSITY OF SYDNEY

Abstract

Faculty of Engineering & Information Technologies
School of Aerospace, Mechanical, and Mechatronic Engineering

Bachelor of Engineering (Mechatronic - Space) &
Bachelor of Science (Advanced Mathematics)

by Kelvin Y.S. Hsu

Informative Seafloor Exploration for Benthic Habitat Mapping

While seafloor bathymetry have been mapped extensively over the last few decades, geological and ecological observations of seafloor benthic zones only began recently. Unlike bathymetric mapping, data collection of benthic imagery requires *in situ* exploration - a significantly slower and costly endeavour. An efficient exploration policy would therefore require solving the informative path planning problem. This thesis investigates a Gaussian process based informative exploration policy for benthic habitat mapping.

Contents

List of Figures

List of Tables

Abbreviations

ACFR	Australian Centre of Field Robotics
API	Application Programming Interface
AUV	Autonomous Underwater Vehicle
AVA	All Versus All
BDKD	Big Data Knowledge Discovery
BO	Bayesian Optimisation
CDF	Cumulative Distribution Function
EP	Expectation Propagation
iid	independent and identically distributed
GP	Gaussian Process
GPBC	Gaussian Process Binary Classification
GPC	Gaussian Process Classification
GPLSC	Gaussian Process Least Squares Classifier
GPMC	Gaussian Process Multi-Class Classification
GPR	Gaussian Process Regression
LIDAR	Light Detection And Ranging
LMDE	Linearised Model Differential Entropy
LP	Laplace Approximation
MCJIE	Monte Carlo Joint Information Entropy
MDP	Markov Decision Processes

NICTA	National ICT Australia
OVA	One Versus All
P1NN	Probabilistic One Nearest Neighbor
PDF	Probability Distribution Function
PIE	Predictive Information Entropy
PLS	Probabilistic Least Squares
POMDP	Partially Observable Markov Decision Processes
PRM	Probabilistic Road Map
SBO	Sequantial Bayesian Optimisation
SE	Squared Exponential
SIEF	Science & Industry Endowment Fund
SONAR	Sound Navigation And Ranging

Chapter 1

Introduction

1.1 Motivation

Thanks to optical and acoustic depth sounding technology, detailed ocean terrain maps across a majority of the globe have become increasingly accessible. These information generally takes the form of *Bathymetric* data - recordings of measured depth, slope, rugosity, and similar structural information that summarises the seafloor topography. Currently, bathymetric data has been recorded with advanced techniques such as SONAR (**SO**und **N**avigation **A**nd **R**anging), LIDAR (**LI**ght **D**etection **A**nd **R**anging), and Multibeam Echosounder for more than half a century [Colbo et al., 2014, Niedzielski et al., 2013]. With such volume of bathymetric information, we can reconstruct accurate 3D models for the seafloor terrain through spatial analytics and modeling techniques [Niedzielski et al., 2013].

However, bathymetric data only contains information regarding the spatial structure of the marine terrain. It provides no indication towards the types of marine habitats that resides within parts of the ocean, nor does it contain clues regarding the minerals or natural resources that may be present. Today, less than five percent of seafloor habitats have been explored [NOAA, 2014]. As the ocean covers more than 70% of the globe, this leaves more than 67% of the planet's habitats unexplored despite our deep reliance on much of these undiscovered ecosystems. With big data analysis becoming more feasible in recent years, there has been an increase in scientific and economical demands - from ecologist and geologists to resource and mining industries - for the ability to predict or infer the types of marine habitats or natural resources residing at various marine environments.

Thus, unlike the case for bathymetric data, there is currently a lack of *label* data, which is a summary of the habitats, resources, and other interesting properties observed at various parts of the ocean. This implies the need to map the ocean floor again for label data using vision based sensing equipments. In order to understand the ecological, geological, chemical, and archaeological aspects of the ocean floor, autonomous underwater vehicles (AUVs) are now capable of efficiently collecting information and observations from natural environments of large spatial scale. In the case of benthic habitat mapping, AUVs collect imagery data of seafloor environments, which are then associated with a particular *label* with semantic meaning [Steinberg et al., 2015]. For example, imageries of 'coral' regions receive the label 'coral'. Unfortunately, while bathymetric data can often be measured with decent accuracy at a distance (for example, with SONAR from ships at sea level), such visual imagery can only be obtained through expensive AUV missions that travel deep into the ocean to image underwater environments at a close distance. Together with the immense spatial scale of the benthic seafloor to be explored, this implies that it is impractical to map exhaustively the entire region of interest (ROI) under any reasonable time and cost. Furthermore, AUV missions are limited by power supply, data storage, and computational capabilities [Bender et al., 2013], further limiting the time and hence coverage each AUV mission can achieve.

As such, AUVs must prioritise exploring sub-domains of the ROI that ideally contain the most important and valuable information. This is a form of spatial sampling problem [Rigby et al., 2010], which aims to address the question: given the choice to observe only a few parts of the region of interest, how should one infer the best candidates for observation? AUV missions add another layer of complication to the spatial sampling problem - the candidate locations must form continuous paths that the AUV can physically travel.

This is known as the *informative path planning problem*. The objective of informative path planning is to minimise the overall uncertainty regarding the entire region of interest.

This thesis addresses the informative path planning problem for benthic habitat mapping. There are two main aspects to informative seafloor exploration for which this thesis is concerned with. The first part of this thesis is focused on benthic habitat mapping, where techniques of habitat classification and inference are discussed. The basic theory under which information and uncertainty are measured and quantified are developed and formulated in the general framework, which is

then applied for benthic habitat mapping. The second part of this thesis then proceed to investigate the informative seafloor exploration problem. Using the properties of inference models developed for benthic habitat mapping, a range of path planning policies are discussed and compared. Practical considerations of computational tractability and flexibility then leads to methods the compromises between optimality and feasibility. Finally, this thesis proposes a practical framework for AUVs to autonomously plan informative paths that achieves the highest mapping rate under a classification accuracy criterion.

1.2 Objectives

The high level objective of this thesis is to develop an informative seafloor exploration policy that can produce benthic habitat maps efficiently. Specifically, this thesis focuses on the theoretical and computational aspects of informative path planning that is practical for seafloor mapping. The aim is to address the informative seafloor exploration problem in a principled manner with theoretical grounding, while taking computational feasibility into account.

The goal of this thesis can be summarised as follows:

> To investigate informative seafloor exploration policies for an AUV with limited mission time, in order to map benthic habitats faster in a principled and computationally feasible way.

1.3 Contribution

This thesis is concerned with mapping seafloor benthic habitats efficiently in a principled manner. Specific contributions of this thesis are:

- A computationally efficient, parallelisable implementation for multiclass Gaussian process (GP) classifiers. Time and spatial complexity of the classifier are further improved through taking advantage of the GP classifier structure under Laplace approximation.
- An alternative, analytically tractable entropy measure hereby named *linearised differential entropy* (LDE). The linearised differential entropy of GP classifiers captures mutual information and uncertainty of a given region of interest. Specifically, instead of examining uncertainty with mainly prediction variance, linearised differential entropy also takes into account the prediction bias. Through addressing the bias-variance trade-off, a common challenge in machine learning algorithms, linearised differential entropy provides an alternative way to quantify uncertainty and information. Analytical tractability is then achieved through linearisation approximations. Linearised differential entropy is proposed as an acquisition function for informative path planning, hereby referred to as *LDE acquisition*.
- A receding horizon framework to informative path planning. This aims to provide a framework that is computationally efficient yet produce informative paths that are stable and near optimal. Together with LDE acquisition, the suitability of the receding horizon approach is demonstrated on the Scott Reef data set from IMOS [2009]. This provides a computationally efficient approach to path planning that is able to run on an AUV in an online manner with suitable hardware. This thesis demonstrates that under a receding horizon framework, acquisition under linearised differential entropy achieves a faster mapping rate than traditional Monte Carlo approaches, and does so with less computational time.

1.4 Nomenclature

As a rapidly growing community, machine learning literature often contains multiple naming conventions and acronyms for similar concepts. Furthermore, in this thesis, new nomenclatures are introduced for concise reference of the concepts developed for the work presented. This sections summarises the major naming conventions employed in this thesis, as well as the nomenclature unique to and introduced by this work.

1.4.1 Naming Conventions

Predictive Information Entropy

Marginalised Predictive Information Entropy

Monte Carlo estimated Joint Predictive Information Entropy

1.4.2 New Nomenclature

Model Differential Entropy/Probabilistic Differential Entropy

Linearised Model Differential Entropy/Linearised Probabilistic Differential Entropy

1.5 Structure

The structure of this thesis is outlined as below.

Chapter 2 provides the necessary background and theory for the purpose of understanding this thesis. Related work in informative seafloor exploration are presented, as well as the various approaches that has been undertaken in this area. Most importantly, this chapter introduces and summarises the Gaussian process framework for both regression and classification. Basic concepts involved in active sampling and informative path planning are then discussed.

Chapter 3 details how the benthic habitat environment are modeled upon bathymetric features using Gaussian process classifiers. A general OVA and AVA framework for multiclass GP classification is developed, which is then applied to test and real datasets to verify performance. The highlight of this chapter is the introduction and derivation of the linearised differential entropy (LDE) for both binary and multiclass GP classifiers. Comparisons with the usual prediction information entropy is presented to demonstrate their differences and respective advantages and disadvantages. This motivates the use of linearised differential entropy as a suitable acquisition function for informative path planning.

Chapter 4 formulates the receding horizon approach to informative path planning, and demonstrate the application of LDE acquisition under this formulation. Description of the simulation experiment are provided, and properties of the approach observed from corresponding results are discussed and analysed. The performance of the proposed exploration policy are then assessed with a classification accuracy criterion.

Finally, chapter 5 summarises the work presented in this thesis, the contributions made, and potentials for future improvements.

Chapter 2

Background

2.1 Related Work

One of the main challenges of informative path planning include selecting the appropriate acquisition function suitable for the type of exploration task at hand. *Acquisition functions*, or *acquisition criterion*, measures the desirability of observing a particular location. In the informative path planning scenario, the acquisition function evaluates the amount of total information or uncertainty contained by a given candidate path. In this context, the way in which *total information* is quantified determines the acquisition function. The more information the path contains, the more desirable it is for the AUV to follow and take observations on that path.

The effectiveness of an acquisition function that measures mutual information throughout any given path has been thoroughly investigated [Bender et al., 2013, Kapoor et al., 2010, Krause et al., 2008, Rigby et al., 2010]. *Mutual information* refers to a measure of information that takes into account the overlapping information within the region or path in consideration. In most spatial sampling contexts, it is unlikely that two given locations within a region or path are completely unrelated or uncorrelated. Observations in one location provides partial information regarding other locations, which is the primary reason that an accurate benthic habitat map can be obtained without visiting all locations within the region exhaustively. Without considering mutual information, an agent such as an AUV may be prompted to observe locations that contain similar information, thus achieving inefficient mapping.

Several types of acquisition functions have been proposed in the benthic habitat mapping context. Bender et al. [2013] uses an acquisition criterion given by (2.1) where π_i^m is the probability of the habitat label being a member of class $m \in \{1, \ldots, c\}$ at query location $i \in \{1, \ldots, n^\star\}$. Here c is the number of classes and $n^\star$ is the number of query points.

$$H = -\frac{1}{n^\star} \sum_{i=1}^{n^\star} \sum_{m=1}^{c} \pi_i^m \log(\pi_i^m) \tag{2.1}$$

For a given query point i, $-\sum_{m=1}^{c} \pi_i^m \log(\pi_i^m)$ is exactly the prediction information entropy (PIE) at that point, which captures the local model uncertainty and thus potential information. As a result, the acquisition criterion given in (2.1) is the mean of the marginalised PIE across query points, which does not capture mutual information through considering the joint distributions of the class labels across multiple query points $i \in \{1, \ldots, n^\star\}$. We will refer to this acquisition criterion as MIE for mean of information entropy. In this approach, the quality of the habitat models are assessed by utility functions that evaluate the confidence of the GP classifier prediction, which is the difference of the MIE before and after an observation is made (2.2) [Rigby et al., 2010]. This utility is used in conjunction with a GP classifier under probabilistic least squares approximation to optimise selected survey paths through considering the differential entropy across the entire region of interest (ROI). Below, we define $X, \mathbf{y}$ to be the observed feature locations and habitat labels, $X^\star$ to be the unobserved query locations, and $X_p, \mathbf{y}_p$ to be the observations made after traversing the proposed path.

$$I = H[X^\star | X, \mathbf{y}] - H[X^\star | X \cup X_p, \mathbf{y} \cup \mathbf{y}_p] \tag{2.2}$$

Krause et al. [2008] instead focuses on sensor placement methodologies to optimse mutual information gained. This aims to reduce predictive variance at all query points. The chosen acquisition function is the difference between the MIE at all *final* unobserved locations $X^\star \backslash X_p$ before and after the observations are taken (2.3).

$$I = H[X^\star \backslash X_p | X, \mathbf{y}] - H[X^\star \backslash X_p | X \cup X_p, \mathbf{y} \cup \mathbf{y}_p] \tag{2.3}$$

Under the GP classification model, however, this results in an integral that can only be estimated through sampling techniques which are computationally expensive. As such, computationally tractable methods usually employ experimental design philosophies in minimising the predictive variance [Bender et al., 2013].

Kapoor et al. [2010] demonstrates the use of posterior mean and covariance of the GP classifier latent function. While this approach takes full advantage of the analytical forms available for a Gaussian latent process, it does not consider the mutual behaviour of the unobserved locations with the proposed path.

On the other hand, Marchant et al. [2014] approaches the problem from a continuous path-planning perspective for UAVs in which two Gaussian processes are used - one to model the phenomenon and another to assess the quality of proposed paths. However, this layered Sequential Bayesian Optimisation approach is primarily devised for a regression setting which becomes intractable when performed in the classification domain.

Much of the intractability discussed in such literature can trace its reason down to the fact that benthic habitat mapping is a classification problem instead of a regression problem. While Gaussian process regression provide analytical forms for inference, due to a non-Gaussian likelihood response, Gaussian process classification is no longer analytically tractable. Instead, several approximations must be made. Four of the most popular approximations to GP classification, in increasing accuracy and computational complexity, are Probabilistic Least Squares, Laplace Approximation, Expectation Propagation, and Variational Inference [Rasmussen and Williams, 2006].

In this paper, we employ Laplace approximation with a probit likelihood response, and further address the tractability status of the technique through proposing an alternative acquisition criterion that captures mutual information. In the current literature, acquisition criterion either do not directly capture mutual information, such as (2.1), or take too long to compute due to required estimations from sampling, such as (2.3). We show that linearised differential entropy (LDE) as an acquisition function captures an appropriate form of mutual information. Specifically, it regularises between reducing mutual bias and mutual uncertainty through taking advantage of both the latent mean and covariance structure, as well as the non-Gaussian likelihood response. Tractability of the approach is further improved through a linearisation approximation on such an acquisition criterion.

2.2 Gaussian Processes

Gaussian processes (GP) are stochastic processes which generalises the multivariate Gaussian distribution. In a statistical learning and machine learning context, they are categorised as a type of *supervised learning* method, which describes the problem of learning relationships between input and output variables from empirical data. The empirical data is also often referred to as the training set.

Supervised learning methods are often further categorised into regression and classification problems, depending on the nature of the output variable. The problem is a regression problem if the output is continuous, and a classification problem if the output is discrete. In the ocean environment modeling setting, bathymetric modeling is a regression problem (when feature extraction is infeasible), and environment type prediction is a classification problem.

In both regression and classification settings, the input variables are often also referred to as *features*, which motivated the term "bathymetric features" in the previous sections - especially to distinguish it from the spatial inputs. In statistical literature, continuous regression outputs are sometimes called *response* variables, although it is more often simply referred as the *output* or *target* in the machine learning community. Discrete classification outputs are referred to as *labels*. In this context, *environment type* and *environment labels* are synonymous.

In the sections which references the use of Gaussian process models, $\mathbf{x}$ will denote the input variable or features of the problem while y will denote the output or target variable. Note that in general there are multiple features such that the input is a feature vector $\mathbf{x}$. Without loss of generality, however, the output variable can always be treated as a scalar quantity y. Under cases of multiple output variables, the problem can be split into multiple single output variable problems. It is true that prediction performance can actually be improved by considering the output vector together, which leads to multi-task regression, as will be briefly discussed. Nevertheless, the main bulk of the content revolve around single output Gaussian processes.

The work presented here will be primarily based on Rasmussen and William's work in *Gaussian Process for Machine Learning* from 2006 [Rasmussen and Williams, 2006].

2.2.1 Bayesian Modeling with Gaussian Processes

The Gaussian process formulation follows the Bayesian modeling philosophy. An important distinction Bayesian modeling makes from the classical approach is the idea of estimating a distribution instead of a point value. While this is often more computationally expensive, it provides a very robust and accurate framework for prediction and analysis. More importantly, it provides capabilities that classical approaches do not possess - the ability to quantify prediction uncertainties.

The basic Bayesian modeling process begins with a prior distribution $p(H)$, the probability distribution of prediction model H being representative, and updates this to a posterior distribution $p(H|D)$, the updated probability distribution of prediction model H being representative after observing a particular data set D [1].

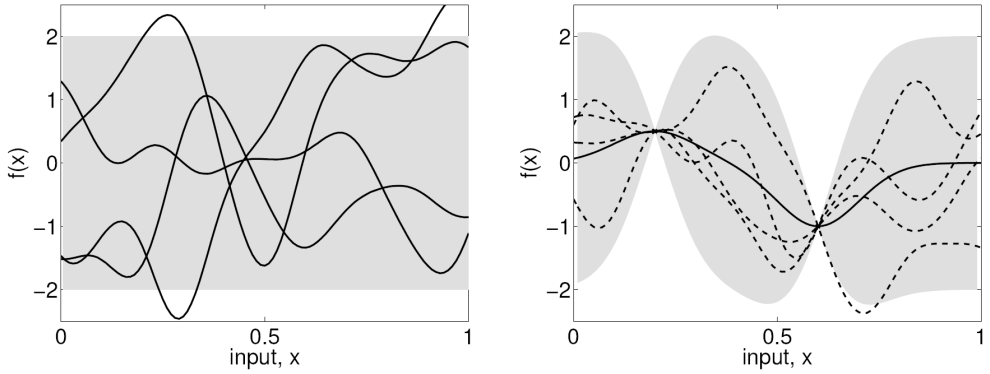
This procedure is illustrated in figure 2.1, where the posterior is updated by two observations. This example further serves to illustrate the concept of distributions over functions, which behaves as an infinite-dimensional generalisation of a multi-variate probability distribution. Instead of drawing random finite vectors from distributions, a random function is drawn from a *process*, a term used to refer to infinite [2] dimensional multi-variate distributions. It is helpful to conceptualise functions as an infinite string of points, such that drawing from an infinite dimensional distribution is equivalent to drawing from processes that operate on function space.

From this illustration, there are a few qualities one can notice. Firstly, the prior distribution is simply the zero function. The prior is meant to represent the system's current belief before the next observations are to be made. In this case, the prior situation involves no observations at all. Ideally, this means that the prior distribution should contain no predictive information. However, it is of philosophical note that all informative [3] inferences must start off with some assumptions regarding the structure of this problem. In the illustration above, the function is assumed to be distributed as a process with zero mean. This assumption has

[1] This discussion employs the common notational convention that the event of observing D is also named D, and the event of model H being the most representative is also named H.

[2] When the input variable is temporal, a process can be more realistically interpreted as having indefinite dimensional distributions.

[3] It is certainly possible to perform inference without any assumptions. It will simply be uninformative in prediction or result.



(a), prior (b), posterior

FIGURE 2.1: Illustration of Gaussian Process Bayesian Modeling: The mean prediction is shown as the solid line. Four samples are drawn in each case, represented by the dashed line. The shared region represents the 2-σ bounds of the prediction at each input feature value x. Two points are observed which updated the distribution from the prior to the posterior.
Figure (a) shows the situation for the prior distribution.
Figure (b) shows the situation for the posterior distribution.
Source: Rasmussen and Williams (2006) [Rasmussen and Williams, 2006]

excluded processes without means, such as the Cauchy process, as well as assumed a rather arbitrary mean function. However, this assumption is often valid as one can always pre-process the output data set through subtracting off their empirical mean so that the output is distributed about zero. The representation of the variance functions as confidence bounds centered at the mean function also hints that multi-model processes are excluded. In fact, the illustration shows a Gaussian process, which is indeed uni-modal - just as its finite dimensional form.

Speaking of the variance, a second observation is that the standard deviation and hence variance of the output function decreases at the observations, and gradually increases away from the observations. This leads to two remarks. Firstly, the variance or uncertainty of the output function at a location reduces when observations are made at that location. Secondly, neighboring points are related - the closer they are, the more related they are. This is seen through the observed points dragging nearby points towards it while reducing the uncertainty of nearby points. This resembles the concept of covariance. Evidently, points closer to each other have higher covariance then those further away, and the covariance beween the same points simply become its variance.

The two observations above demonstrate that, just as a Gaussian distribution is defined through a mean vector and a covariance matrix, a Gaussian process

is defined through a mean function $m(x)$ and a covariance function $k(x, x')$. In Gaussian process literature, the covariance function is also called a *kernel* function.

Formally, a function $f(x)$ is distributed as a Gaussian process $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$ if for any finite collection of points $\mathbf{x} := [x_1, x_2, \ldots, x_n]^T$, the corresponding output vector $f(\mathbf{x}) := [f(x_1), f(x_2), \ldots, f(x_n)]^T$ is jointly distributed as a multi-variate Gaussian such that $f(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$. Certainly, this is true for finite dimensional multi-variate Gaussian distributions, in that any subset of a said distributed random vector will also be multi-variate Gaussian distributed of lower dimensionality. The next few sections will discuss the kernel matrix K more rigorously.

Finally, as mentioned before, the mean function can be generally assumed to be the zero function, as at each stage of inference both the model and data can be subtracted by their theoretical or empirical means. This elucidates that Gaussian processes are completely defined by their kernel function k.

2.2.2 Kernel Functions

2.2.2.1 Stationary Kernels

As kernel functions completely define the prediction characteristics of a Gaussian process, this section aims to provide the mathematical background regarding kernels that are necessary for understanding Gaussian processes. The following discussion will only cover the minimal background necessary for further sections, as treatments of kernel functions can easily become very detailed and rigorous once analysis begins in topics such as differentiability effects or eigenfunction decomposition. Further treatment of this material is available through Rasmussen and Williams (2006) [Rasmussen and Williams, 2006].

Intuitively, the kernel function determines the *similarity* between data points. This is a notion that all supervised learning algorithms intend to do, although rather implicitly in most cases. The $\mathcal{GP}$ formulation makes this explicit through the covariance between any two points in the feature space.

Kernel functions can be categorised into stationary kernels and non-stationary kernels. In this thesis, non-stationary kernels will become of vital importance in bathymetric and environment modeling.

Stationary kernels are ones whose covariance properties do not depend explicitly on the locations $\mathbf{x}$ and $\mathbf{x}'$ of consideration, but only on the difference $\mathbf{x} - \mathbf{x}'$ between them. Thus, the covariance properties are *stationary*, or invariant, under translations in the feature space.

Common stationary kernels are the squared exponential kernel [4] and the Matérn kernels. The squared exponential (SE) kernel between any two points $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^m$ in the feature space with m features has the following form (2.4).

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\Big(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}') \Big) = \sigma_f \exp\Big(-\frac{1}{2} a^2 \Big)$$
$$\Sigma = \begin{bmatrix} l_1^2 & l_{12} & \dots & l_{1m} \\ l_{21}^2 & l_2^2 & \dots & l_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1}^2 & l_{m2} & \dots & l_m^2 \end{bmatrix} \tag{2.4}$$

Here, σ_f is called the sensitivity, and determines the overall reference strength scale of the covariance function. The matrix Σ is the length scale matrix, and determines the reference length scale and principle axis directions within the feature space. Like most quadratic forms, Σ is required to be symmetric and positive semi-definite. In particular, when Σ is diagonal, the kernel is termed *axis aligned*. When Σ is proportional to an identity such that $\Sigma = l^2 I_{m \times m}$, the kernel is termed *isotropic*.

The sensitivity parameter σ_f and length scale parameters l_{ij}, $i, j \in 1, 2, \dots, m$ with $l_i := l_{ii}$ completely withhold the information of a squared exponential kernel. Unlike parametric models, however, while these parameters define the kernel directly, they define the $\mathcal{GP}$ model indirectly. Because of the multiple levels of relation from these parameters to the model, these parameters are termed *hyperparameters* of the $\mathcal{GP}$.

In practice, it is often possible to pre-process the data or transform the feature space so that an axis aligned kernel can be applied. The assumption imposed is that the principle axis directions are aligned with the feature space axis. In

[4] Squared exponential kernels are also sometimes called Gaussian kernels. However, in conversations it tends to create confusion between the probability density function $\phi(x)$ for Gaussian distributions and the covariance function $k(x, x')$ itself, so this term is avoided in this thesis.

this case, the $\mathcal{GP}$ model is defined by $m+1$ hyperparameters, where m is the number of features. Even without the axis-aligned assumption, the number of hyperparameters is $\frac{m(m+1)}{2}+1$.

The above formulation suggests to define $a^2 := (\mathbf{x}-\mathbf{x}')^T\Sigma^{-1}(\mathbf{x}-\mathbf{x}')$. This can be interpreted as the squared distance between $\mathbf{x}$ and $\mathbf{x}'$ under the warp defined by Σ. In particular, when the feature space is isotropic such that $\Sigma = l^2 I_{m\times m}$, then $a = \frac{r}{l}$ where $r := +\sqrt{r^2}$, $l := +\sqrt{l^2}$, and $r^2 = (\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')$. In fact, most stationary kernels are functions solely of a^2, or with the definition $a := +\sqrt{a^2}$, they are simply scalar functions with scalar inputs $a = a(\mathbf{x}, \mathbf{x}')$. This form also makes evident that the covariances between two points decreases monotonically as the distance between them increases - a property most kernel functions exhibit.

Continuing with this formulation, the Matérn class of kernel functions are given by (2.5).

$$\begin{gathered} k_{\text{Matérn}}(\mathbf{x},\mathbf{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\Big(\sqrt{2\nu}a\Big)^{\nu} K_\nu\Big(\sqrt{2\nu}a\Big) \\ a^2 := (\mathbf{x}-\mathbf{x}')^T\Sigma^{-1}(\mathbf{x}-\mathbf{x}') \end{gathered} \tag{2.5}$$

Γ and K_ν are the Gamma function and modified Bessel function respectively [Rasmussen and Williams, 2006]. ν is a positive hyperparameter that determines the differentiability property of the Matérn class kernel. The $\mathcal{GP}$ model with Matérn class kernel is d-times mean square differentiable if and only if $\nu > k$. In the limit of $\nu \to \infty$ for infinite differentiability, the Matérn kernel becomes the squared exponential kernel (2.4). While the general Matérn class kernel seem complicated due to the Gamma function and modified Bessel function, its form become simple for $\nu = p + \frac{1}{2}$ where p is a non-negative integer. That is, Matérn kernels with $\nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}, \ldots$ have simple analytic forms without reference to the modified Bessel function. In fact, for $\nu > \frac{5}{2}$, the degree for which the Matérn kernel changes becomes quite unnoticeable for most practical purposes such that it may as well be replaced by the squared exponential kernel with $\nu \to \infty$. Similarly, while there is a more noticeable effect of changing ν within the range $\nu \in (0, \frac{5}{2})$, in practice is it is often not worth the expense of implementing the complicated form for a almost unnoticeable improvement in modeling accuracy. Hence, it is replaced with the Matérn kernel $\nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}$, whichever is the closest. In this way, in practice only the Matérn kernels with $\nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}$ are employed, and they are respectively

termed the Matérn 1/2 kernel, Matérn 3/2 kernel, and Matérn 5/2 kernel - in the order of increasing differentiability. These kernels have forms as listed below (2.6).

$$
\begin{aligned}
k_{\text{Matérn, } \nu=\frac{1}{2}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \exp(-a) \\
k_{\text{Matérn, } \nu=\frac{3}{2}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 (1 + \sqrt{3}a) \exp(-\sqrt{3}a) \\
k_{\text{Matérn, } \nu=\frac{5}{2}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \Big(1 + \sqrt{5}a + \frac{5}{3}a^2\Big) \exp(-\sqrt{5}a) \\
a^2 &:= (\mathbf{x} - \mathbf{x}')^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}')
\end{aligned}
\tag{2.6}
$$

Together, the squared exponential kernel and the Matérn class kernels provide a flexible set of kernel functions that can model a multitude of phenomena from various fields such as geology, ecology, finance, logistics, control theory, and machine learning. Specifically, kernel functions make spatial relations explicit which assists in the interpretation in spatial modeling cases, including bathymetric modeling and ocean environment modeling.

2.2.2.2 Non-Stationary Kernels

Non-stationary kernels introduce flexibility for modeling phenomenons where the inherent length scales varies across feature locations. The problem with stationary kernels is that the GP will always learn length scales that are as small as it needs to for modeling the fastest varying phenomenon in the model. While the marginal likelihood inherently balances modeling accuracy and overfitting, when it comes to the choice between modeling a peak in data variation with a risk of overfitting the rest of the data or ignoring that peak, the optimiser will always prefer the former as marginal likelihood gain from successful modeling is higher than loss from overfitting. Because learning stage is done through optimising the marginal likelihood, this forces the length scale to be smaller than it needs at slower varying places.

Figure 2.2 illustrates the non-stationary Gaussian process for a terrain modeling application. Note that this does not imply that it is only relevant for GP regression problems - the latent functions used in GP classification is itself a GP regression problem for which length scale interpretation is almost identical to that shown in figure 2.2.

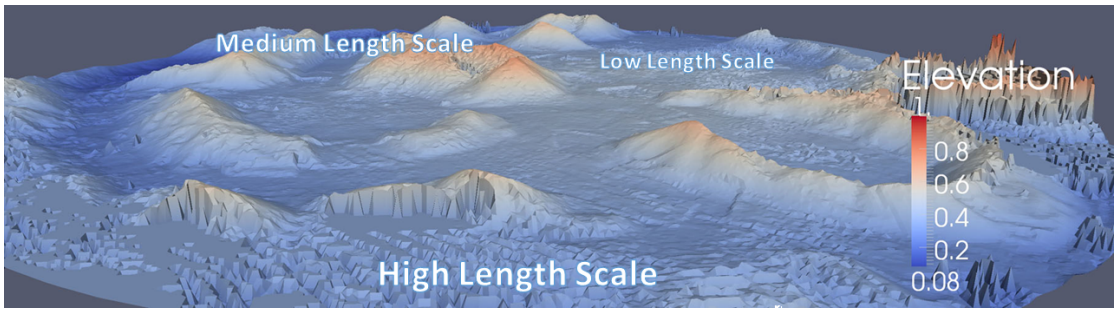


FIGURE 2.2: Illustration of non-stationary Gaussian process for terrain modeling [Tong, 2013]
Flat parts have high length scales (slow varying) while rough parts have low length scales (fast varying)

The non-stationary kernel function employed is the Paciorek Non-Stationary Covariance Function (2.7) [Lang et al., 2007].

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 |\Sigma_i|^{\frac{1}{4}} |\Sigma_j|^{\frac{1}{4}} \left| \frac{\Sigma_i + \Sigma_j}{2} \right|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j) \left(\frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right] \tag{2.7}$$

The matrices Σ_i and Σ_j are the local length scale matrices at $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively, and are interpreted the same way as the stationary case. The only difference is that these length scale matrices only operate locally, and are functions of the input feature vector $\mathbf{x}$. In each kernel location, two length scale matrices are queried. Hence, in a kernel matrix of size $n \times m$, around $n + m$ unique queries are made if no feature locations overlap.

It is worthwhile to observe that the effective length scale matrix in the exponent is the average of the two length scale matrices, with its effect reduced with increasing distance between the two points of consideration as with all kernels.

Note that the normalisation matrix determinants are chosen such that the kernel function is reduced into a stationary one when $\Sigma_i = \Sigma_j = \Sigma$ (2.8).

$$
\begin{aligned}
k(\mathbf{x}_i, \mathbf{x}_j) &= \sigma_f^2 |\Sigma|^{\frac{1}{4}} |\Sigma|^{\frac{1}{4}} \left| \frac{\Sigma + \Sigma}{2} \right|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j) \left(\frac{\Sigma + \Sigma}{2} \right)^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right] \\
k(\mathbf{x}_i, \mathbf{x}_j) &= \sigma_f^2 |\Sigma|^{\frac{1}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j) (\Sigma)^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right] \\
k(\mathbf{x}_i, \mathbf{x}_j) &= \sigma_f^2 \exp \left[-\frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j) \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right]
\end{aligned}
\tag{2.8}
$$

That is, the Paciorek non-stationary kernel reduces to the squared exponential kernel under the stationary limit. In this way, the Paciorek kernel generalises the squared exponential kernel.

2.2.3 Regression

Gaussian process regression is a regression technique that employs Gaussian processes as its inference model. Because Gaussian processes already operate on function spaces with continuous inputs and outputs, no extra pre-processing or transformations are needed. The bulk of the technique thus lies in learning the kernel function of the Gaussian process. Gaussian process regression is also called *kriging* or Kolmogorov Wiener prediction when used for interpolating geospatial data in a geostatistics setting. This section attempts to summarise the important concepts regarding $\mathcal{GP}$ regression and how they work. More rigorous discussions are available from Rasmussen and Williams (2006) [Rasmussen and Williams, 2006].

Once a kernel function is chosen, such as the squared exponential or Matérn kernels, learning the kernel function becomes equivalent to learning the hyperparameters of the kernel. In this way, the Gaussian process model is actually defined completely by its hyperparameters, which are often only handful in quantity. This illustrates that while its temporal complexity $\mathcal{O}(n^3)$ is quite high, its spatial complexity and memory requirements are quite moderate at $\frac{m(m+1)}{2} + 1 + n(m+1)$ real numbers, where $n(m+1)$ real numbers comes from the training data itself.

With a given kernel function k as the covariance function, by definition we have that the training observations $\mathbf{f}$ and the query observations $\mathbf{f}^\star$ is distributed as a multivariate Gaussian (2.9).

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^\star \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X^\star) \\ K(X^\star,X) & K(X^\star,X^\star) \end{bmatrix}\right) \tag{2.9}$$

where the matrix $K(X, X')$ is defined to have elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}'_j)$ with X and X' defined as the canonical data design matrix form (2.10), both of each can take either the matrix of training points X of size n or query points $X^\star$ of size $n^\star$. To shorten notation, it is customary to define $K := K(X, X)$, $K^\star := K(X, X^\star)$, $K^{\star\star} := K(X^\star, X^\star)$, where the symmetry of the covariance matrix readily yields $K^{\star T} := K(X^\star, X)$. Specifically, K will be referred to as the data kernel.

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \cdots \\ \mathbf{x}_n^T \end{bmatrix} \qquad X' = \begin{bmatrix} \mathbf{x}'^T_1 \\ \mathbf{x}'^T_2 \\ \cdots \\ \mathbf{x}'^T_n \end{bmatrix} \tag{2.10}$$

The joint distribution readily contains information for the conditional distribution of the query points given the training points $p(\mathbf{f}^\star|\mathbf{f})$ knowing the training points and query locations [5]. This leads to the posterior distribution (2.11).

$$\mathbf{f}^\star|\mathbf{f} \sim \mathcal{N}(K^{\star T}K^{-1}\mathbf{f}, K^{\star\star} - K^{\star T}K^{-1}K^\star) \tag{2.11}$$

A comparison with the prior $\mathbf{f}^\star \sim \mathcal{N}(\mathbf{0}, K^{\star\star})$ shows the mean effect $K^{\star T}K^{-1}\mathbf{f}$ and covariance effect $-K^{\star T}K^{-1}K^\star$ which introduces observed information into the model. Interestingly, as $K^{\star T}K^{-1}K^\star$ is positive definite, this intuitively means that the observation has reduced uncertainty in the model.

The above posterior formulation encompasses the heart of the GP regression model. The rest of the discussion will focus on detailed aspects of its implementation and variants.

Under noisy observations, a hyperparameter σ is introduced for the standard deviation of the noise, which is also assumed to be *iid* and Gaussian distributed with standard deviation σ and zero mean. The only alteration is the observations are notated as $\mathbf{y}$ instead of $\mathbf{f}$, and most importantly the data kernel is to be replaced

[5] Throughout this thesis, since the feature locations X and $X^\star$ are always known, they are inherently conditioned upon and will not be shown explicitly in notation.

with $K \mapsto K + \sigma^2 I$. Note that however the query points remain as the latent function $\mathbf{f}^\star$. If one were to predict future observations $\mathbf{y}^\star$, then it suffices to generate $\mathbf{f}^\star$ from the posterior and add randomly generated noise with standard deviation σ.

2.2.3.1 Hyperparameter Learning

One of the most important yet tricky aspects of GP modeling is the training stage. Since the model is determined entirely by the hyperparameters, the hyperparameters must be optimised in accordance to some fitness metric. The fitness metric employed to be maximised is the marginal likelihood, otherwise termed as evidence, of the observed data. This is usually non-trivial to calculate and, in most cases, analytical forms do not exist. Fortunately, due to the Gaussian assumption of he GP model, there exists an analytical form for the marginal likelihood. In practice, however, it is computationally faster to compute the log marginal likelihood (2.12).

$$\log(p(\mathbf{f})) = -\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f} - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi) \tag{2.12}$$

Again, with noisy observations, corresponding substitutions with the data kernel matrix $K \mapsto K + \sigma^2 I$ and observations $\mathbf{f} \mapsto \mathbf{y}$ is to be made.

The last term is a constant, so it can be ignored during the optimisation stage and included back in once optimisation completes.

In practice, hyperparameter learning can be sped up by employing the fact that $\frac{1}{2}\log|K| = \sum_i L_{ii} = \text{trace}(L)$ where L is the Cholesky decomposition of K (or $K + \sigma^2 I$ in the noisy case). In fact, the Cholesky decomposition L leads to better numerical stability when inverting the matrix K since $K = LL^T$ so that $K\backslash\mathbf{y} = L^T\backslash(L\backslash\mathbf{y})$.

2.2.3.2 Sampling from a Gaussian Process

2.2.4 Classification

The GP classification method is of vital importance to the environment modeling process. It is used to distinguish and predict the environment type with a measure of entropy in order to quantify the information reward one can gain by exploring the area.

Unlike the regression case, because the output labels are no longer continuous, it cannot be represented by a continuous probability density function such as a Gaussian. As such, a continuous latent function is introduced in the GP classification process. Intuitively, this latent function quantifies and measures the distinct qualities of the label. As a binary classification example, if the classifier is to distinguish between "Apples" and "Oranges", the latent function would then represent the "Appleness" of each observation, with high "Appleness" corresponding to observations likely to be "Apples" and low "Appleness" corresponding to observations likely to be "Oranges". Note that only one latent function is needed in the binary case. The latent function is then "squashed" into the unit range [0, 1] so that it can be interpreted as a probability.

Although the latent function is now continuous such that it is possible to model it with a GP regression model, the posterior probability is in general non-Gaussian distributed. In this way, approximations are necessary.

There exists four approximate methods for GP classification. In increasing orders of accuracy and decreasing order of implementation difficulty, they are Probabilistic One Nearest Neighbor (P1NN), Laplace Approximation (LP), Expectation Propagation (EP), and Gaussian Process Least Squares Classifier (GPLSC) [Rasmussen and Williams, 2006]. In this section, Laplace approximation is chosen as a reasonable balance between accuracy and implementation difficulty.

2.2.4.1 Response Functions

Before discussions begin regarding the various types of GP classifiers, it is important to understand the role of *response functions* in Bayesian classifiers.

The response function is sometimes also called a sigmoid function. These functions must satisfy the requirement that is it monotonically non-decreasing with a domain of all real numbers $\mathbb{R}$ and a range of unit interval $[0, 1]$. That is, $\lambda(z) : \mathbb{R} \mapsto [0, 1]$. As referenced above, these functions serve to "squeeze" the latent functions into a range where probabilistic interpretation is possible.

The most widely used response function is with no doubt the logistic function (2.13). Response functions that are also commonly used in a GP classifier setting include the normal cumulative distribution function (2.14).

$$\lambda(z) = \frac{1}{1 + \exp(-z)} \tag{2.13}$$

$$\lambda(z) = \Phi(z) := \int_{-\infty}^{z} \phi(x)dx = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}x^2\Big)dx \tag{2.14}$$

2.2.4.2 Binary Classification

The Laplace approximation for GP binary classification works by learning a latent function through an iterative scheme involving successive GP regressions, and "squeezing" that latent function into an appropriate probabilistic range for likelihood interpretation. The Laplace approximation approaches the approximation problem by determining the mean and variance of the true distribution and approximating the said distribution with a Gaussian distribution of the same mean and variance. Assuming Gaussian distributions, the prior and the evidence (marginal likelihood) have analytical forms. Under such assumptions it is therefore possible to marginalise out the likelihood predictions to obtain a posterior estimate. Finally, this learning stage can be trained by optimising the marginal likelihood [6].

[6] At this stage of the thesis progress, it is uncertain whether or not a deeper and more rigorous theoretical grounding should be provided for GP classification. Perhaps it would benefit by only providing an intuitive outline of the procedure, rather than a rigorous derivation, so that the reader is not distracted from the main contributions of this thesis later on.

2.2.4.3 Laplace Approximation

2.2.4.4 Probabilistic Least Squares

2.2.4.5 Multiclass Classification: One v.s. All

While a consistent framework for the GP Multi-class classification using Laplace approximation exist, a simpler and perhaps a more computationally efficient approach is to employ the One v.s. All (OVA) and All v.s. All (AVA) philosophy. The following discussion assumes that there are $c > 2$ classes of labels for which classification is to occur.

The OVA approach performs classification by introducing c *independent* classification problems, each trying to classify a label against all others. Each of these problems are thus a binary classification problem for which solution methods are known. Once the predictive probabilities from all learned GP classifiers are obtained, a consistent framework is used for fusing the separate prediction probabilities into a coherent prediction probability. This is necessary as the binary classifiers are independently learned and performs prediction independently, and may not necessarily provide coherent results. In fact, simply stacking the prediction probabilities together yields a "probability" distribution that does not sum up to 1.

The AVA approach operates similarly. It insteads introduces $\frac{c(c-1)}{2}$ *independent* classification problems, each classifying between a pair of labels. The same exact philosophy follows in that a final consistent framework is needed for fusing the prediction probabilities into one coherent prediction probability. It is often more difficult for probability fusion in the AVA setting as compared to the OVA setting.

2.2.4.6 Multiclass Classification: All v.s. All

2.2.4.7 Fusion of Prediction Probability

This section should be in the GP implementation chapter of this thesis (e.g. Chapter 3). It would further the discussion above regarding the probability fusion problem by introducing the exclusion method and mode keeping method that has been

developed and tested. It is anticipated that other methods will be developed by then, so that this section would undergo significant editing.

2.2.4.8 Entropy

After a predictive probability distribution is obtained for each query environment location, the uncertainty of such predictions can be quantified through the entropy of of that distribution.

The output of a GP classifier is a discrete probability distribution with finite size - equal to the number of classes. For a general discrete probability distribution $p(x)$, the entropy is quantified as below (2.15).

$$H(p(x)) = -\sum_i p(x_i)\log(p(x_i)) \tag{2.15}$$

Although it is unlikely that the entropy of bathymetric feature modeling is required, the regression posterior is a continuous probability density $p(x)$ for which a similar form for distribution entropy exists (2.16).

$$H(p(x)) = -\int_{\Omega_x} p(x)\log(p(x))dx \tag{2.16}$$

With these entropy measures, predictive uncertainties can be quantified, which allows active information seeking plans to be possible.

2.2.4.9 Sampling from a Gaussian Process for Classifiers

2.3 Active Sampling

2.3.1 Static Active Sampling

2.3.2 Dynamic Active Sampling

2.4 Informative Path Planning

Path planning under dynamic uncertainty has been a challenging task for all information searching missions. This class of path planning problems have the special property that there is no goal location, and no stationary node, edge, or field cost to be cumulatively minimised. The objective is to reduce the overall uncertainty or entropy of a particular region given indefinite time. The complication is introduced with the non-linear dynamics of the uncertainty or entropy field of the region each time a planned path is executed, which also makes the solution extremely frequency dependent.

Prior work and attempts at the active path planning problem include Marchant and Ramos (2014), where Bayesian Optimisation (BO) techniques combined with Gaussian process models are employed in an environmental monitoring setting [**?**]. In this layered Bayesian Optimisation approach, two Gaussian processes are used - one to model the phenomenon and the other to model the quality of selected paths. Through Bayesian optimisation, sampling over continuous paths are optimised which maximises the reward over the final mission trajectory. The path planning process is done using Markov Decision Processes (MDP) with a Reinforcement Learning approach. Rapidly Exploring Random Graphs (RRGs) is combined with BO to search for informative paths. In this way, a continuous path can be planned through BO [**?**].

This was was further extended by Marchant et. al. (2014) where Sequential Bayesian Optimisation (SBO) is used as online POMDPs for path planning [**?**].

Other prior work includes Brooks et. al. (2006) where the POMDP approach is investigated for continuous state space planning [Brooks et al., 2006]. This

method was compared to previous work with value-based and gradient-based solution methods which seek to transcribe the continuous problem into a discrete problem [7]. One of the most important limitations discussed in this work is that analytical and accurate solutions exist almost only for linear systems with quadratic cost (Linear Quadratic Systems). Otherwise, the other option with non-value based methods require heuristics that can be difficult to justify for its appropriateness to the problem. Nevertheless, through parametrising the problem, parametric POMDPs can provide an accurate solution to the path planning problem under certain assumption such as linear quadratic dynamics [Brooks et al., 2006].

In conclusion, POMDP methods are currently one of the most reliable and accurate method for continuous path planning in an information gathering setting. While the technique enforces limitations on the problem dynamics, with sufficient modeling it is deemed possible to perform path planning to an acceptable level. This thesis will thus investigate POMDP methods for active path planning in the ocean exploration setting.

2.4.1 Myopic and Non-myopic Planning

2.4.2 Advantages of Gaussian Process Models

[7] The author has practiced with transcribing the continuous problem into a discrete problem in a shortest path setting in

Chapter 3

Benthic Habitat Mapping

3.1 Gaussian Process Classifiers for Benthic Habitat Mapping

3.2 Benthic Habitat Modeling

This thesis project can be broken down into two main parts - ocean environment modeling and path planning. Ocean environment modeling itself includes bathymetric feature extraction and modeling, and environment label modeling. This section will focus on the environment modeling aspect.

In order for autonomous underwater vehicles to plan a path that can maximise the amount of information gained regarding a particular ocean region, it would need a method to predict the types of environments it may encounter, with a measure of its prediction uncertainty.

While the path planner is to plan in the spatial space, in general the prediction model operates upon some feature space with more direct and explicit relationships with the output we would like to predict.

It is thus important to make a distinction between the feature space F for which label modeling is to occur and the spatial space S for which bathymetric modeling and planning is to occur. The spatial space usually consists of Cartesian coordinates (x, y) in the eastings-northings frame or the longitude-latitude frame.

The path is to be planned in this spatial space. The frame is usually converted into a local body frame during the execution of control signals for path tracking. However, this does not affect the formulation presented here. The feature space includes bathymetric features for which the labels will be modeled upon. Depending on the features employed for the modeling process, there are usually approximate analytical forms for extracting such features from raw depth observations.

The following features in table 3.1 were chosen by the author for bathymetric modeling.

Feature Name	Feature Symbol	Feature Units
Depth	h	m
Slope (Small Scale)	m_s	m/m
Slope (Large Scale)	m_l	m/m
Rugosity (Small Scale)	r_s	m^2/m^2
Rugosity (Large Scale)	r_l	m^2/m^2

TABLE 3.1: Bathymetric Features

The raw bathymetric data contains depth information at various spatial locations. Such data are often collected rather uniformly in approximate grid formations such that it is possible to calculate the ocean floor slope through finite differencing. The slope is divided into small scale and large scale variations, as marine environments - especially underwater habitats - often depend not only on the immediate slope but also slope variations on the larger scale. The same idea applies to rugosity, the measure of local height variations. The feature extraction process is detailed in section 3.2.2.

Spatial coordinates are chosen to be excluded from the bathymetric feature set. For environment prediction purposes, it is expected that ecological habitats and geological sites exhibit no explicit relationship with the location of the site, and that its properties arise solely due to the local environment characteristics (features) of that site.

Figure 3.1 show a high level overview of the ocean environment modeling process. As bathymetric data is available in more quantities and distributed more uniformly, it is often sufficient to employ the feature extraction process outlined in section 3.2.2 to obtain the bathymetric features for modeling. However, if the bathymetric data is sufficiently sparse or is not distributed uniformly for grid based methods, then the feature extraction process itself becomes a prediction problem.

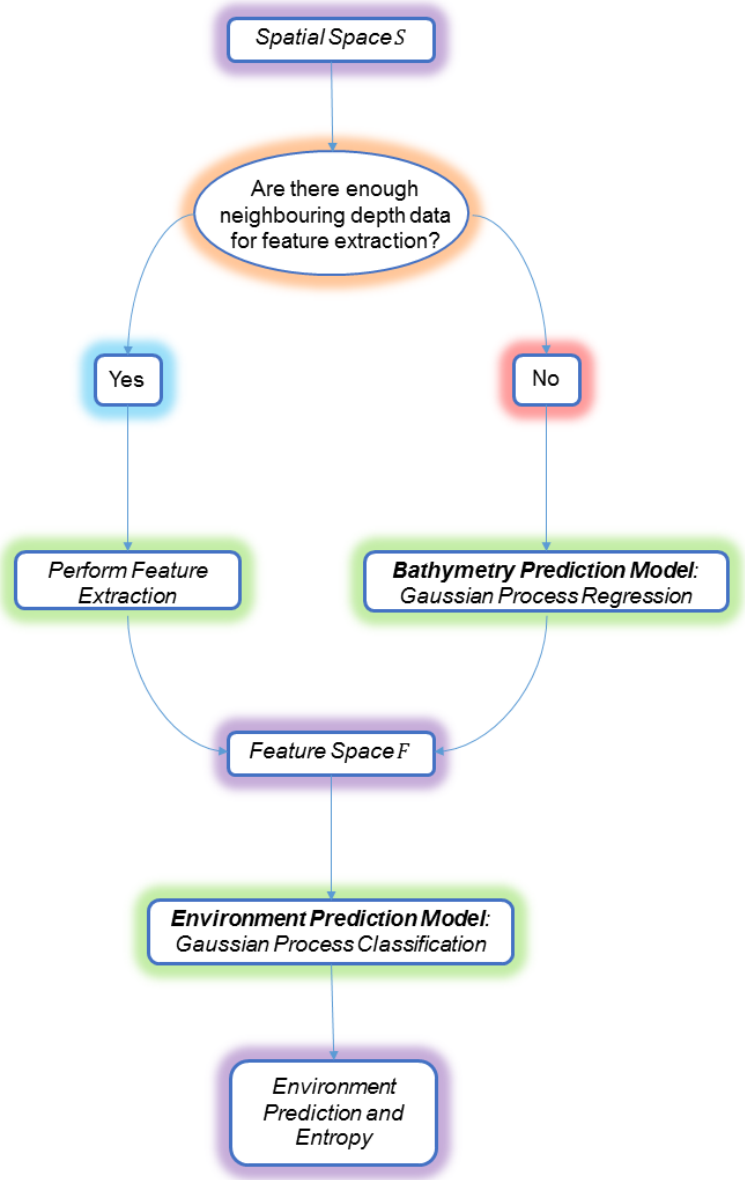


FIGURE 3.1: Environment Modeling

In that case, a Gaussian process regression model is proposed for predicting the features at a given spatial location. While this is much more computationally expensive than performing feature extraction, it is also quite rare that this is necessary under abundant bathymetric data.

Finally, once the feature vectors are obtained at training locations, the environment type is to be predicted. The environment type is summarised through labels that indicate the type of marine environment that was observed or predicted. Common AUV mission examples include "reef", "sand", and "rocks". These labels are often summarised through processing visual and stereo imagery obtained through past AUV missions. With a discrete set of possible labels, the environment prediction problem is to be modeled as a Gaussian process classification problem. From here on, the environment prediction problem is understood to refer to the two stage process of feature extraction or modeling and environment label prediction, with the latter being the main bottleneck for this process.

3.2.1 Data Matching

A subtlety that arises from the above formulation is that during the training stage, the bathymetric data and the label data are not necessarily observed at the same places. Figure 3.2 illustrates the spatial distribution of the two datasets in a typical setting.

While bathymetric data are usually collected rather uniformly, the label data are collected from past AUV missions whose trajectory are continuous curves across the ocean floor [Friedman et al., 2015]. Due to slower AUV velocity as compared to surface ships which often employ SONAR or LIDAR techniques for bathymetry mapping, the label data are also spatially denser and concentrated on the mission trajectory, while being almost non-existent elsewhere.

Therefore, in order to predict label data, the training data would need to be matched accurately. There are two straight forward choices at hand. The first is to estimate the bathymetric features at places where label data exists. However, at places near past mission paths, bathymetric data appears much more sparsely than label data, so that the feature extraction or regression prediction will yield very similar features across manly label data points. This reduces prediction power through a slow varying and limited feature group.

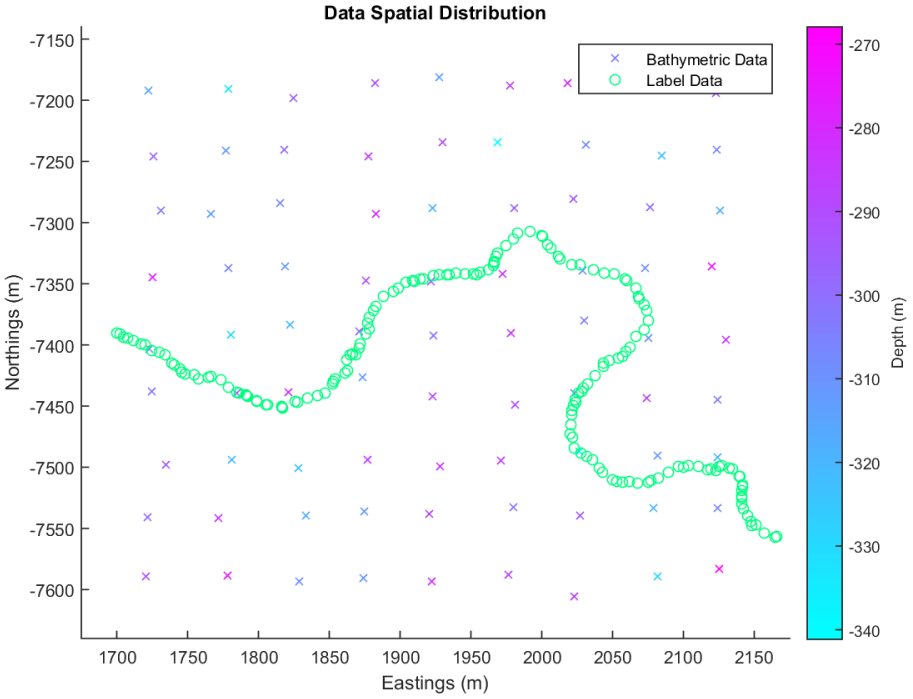


FIGURE 3.2: Illustration of Bathymetric and Label Data Density

Instead, the second choice is to estimate the label data at places where bathymetric data exists. In this setting, regions closer to past mission paths have higher volumes of label data, increasing the amount of training points. Regions further away would naturally generate more prediction uncertainty in the prediction stage.

Hence, second method is chosen to be employed for data matching, in order to form our training set. Naturally, to predict environment labels from bathymetric features, we again need the Gaussian process classification model.

3.2.2 Feature Extraction

The feature extraction process assumes that the bathymetric depth data is available in grid form. That is, one can represent the available depth data $H = \{h_k\}_{k\in 1,2,\ldots,N}$ as $H = \{h_{ij}\}_{i\in 1,2,\ldots,n_i,\ j\in 1,2,\ldots,n_j}$ where varying i and j corresponds to varying data points in axis 1 and 2 respectively. Axis 1 and 2 is required to form an orthonormal frame. While axis 1 and 2 is usually aligned with the eastings-northings frame, it is generally not required for the feature extraction process.

Without loss of generality, let x and y denote quantities corresponding to the orthogonal axes. We have that at (x_i, y_j) $(i \in 1, 2, ..., n_i, \quad j \in 1, 2, ..., n_j)$ the depth is measured as h_{ij}. The partial derivatives of various degrees of accuracy and scale can then be estimated through central differencing, as shown in figure 3.3 [Holoborodko, 2010]. The author has chosen $N = 3$ neighbors for short scale slope and $N = 9$ neighbors for large scale slope.

N	N-point stencil Central Differences
3	$\frac{f_1 - f_{-1}}{2h}$
5	$\frac{f_{-2} - 8f_{-1} + 8f_1 - f_2}{12h}$
7	$\frac{-f_{-3} + 9f_{-2} - 45f_{-1} + 45f_1 - 9f_2 + f_3}{60h}$
9	$\frac{3f_{-4} - 32f_{-3} + 168f_{-2} - 672f_{-1} + 672f_1 - 168f_2 + 32f_3 - 3f_4}{840h}$

FIGURE 3.3: Finite Difference Methods: Central Difference Coefficients
The subscripts i represents

Central differencing is chosen as it is more numerically accurate. The disadvantages of instability and slightly higher time complexity from dynamic cases are not present in the static feature extraction process. Nevertheless, forward differencing is to be used at the boundaries of the dataset where neighboring data is missing on one side.

With two axis, the result is a 2 element gradient vector. It is possible to treat the 2 elements as separate features. However, this would make the modeling problem frame dependent unnecessarily. Therefore, the magnitude of this gradient vector is taken as the slope feature.

Rugosity is a measure of local height variations in the terrain. By definition, its form is computed as $r = A_r/A_g$, the real surface area divided by the geometric surface area.

Under cases without perfect grid formation, such as that shown in figure 3.2, this feature extraction process becomes only an approximation. As the data set deviates from the form assumed above, it can then become necessary to estimate the

features using Gaussian process regression - specifically, the multi-task Gaussian process regression.

On the other hand, under fine-scale bathymetric reconstructions, more sophisticated methods for deriving multi-scale measures of rugosity and slope exist. For example, under bathymetry measurements that are geo-referenced through stereo imagery, rugosity can be calculated through a Delaunay triangulated surface mesh and projecting areas onto the plane of best fit using Principal Component Analysis (PCA) [Friedman et al., 2012].

3.3 GP Classification with Laplace Approximation

3.3.1 Theory

3.3.2 Implementation

3.3.3 Results

3.4 GP Classification with Probabilistic Least Squares Approximation

3.4.1 Theory

3.4.2 Implementation

3.4.3 Results

3.5 Drawing from Gaussian Process Classifiers

3.5.1 Theory

3.5.2 Implementation

3.5.3 Results

3.6 Laplace Approximation and Probabilistic Least Squares

Compare probability outputs, classification outputs, entropy outputs, etc

3.7 The One Versus All Framework

3.8 The All Versus All Framework

3.9 Properties and Performance of OVA and AVA Classifiers

3.10 Probability Fusion Methods

3.10.1 Normalisation Method

3.10.2 Mode Keeping

3.10.3 Exclusion

3.11 Mapping Scott Reef Benthic Habitats

3.11.1 Problems and Solutions to Big Data Analysis

3.11.2 Feature Extraction

3.11.3 Case with 4 Labels

(With different amounts of sampled points)

3.11.4 Case with 17 Labels

Chapter 4

Informative Seafloor Exploration

4.1 Mutual Information

4.2 Monte Carlo Estimated Joint Predictive Information Entropy

Provide pseudocode for limited and good way of doing it

4.2.1 Binary Classification

4.2.2 Multiclass Classification

4.3 Linearised Model Differential Entropy

In this section we introduce the linearised differential entropy of Gaussian process classifiers. This method attempts to address the need for a measure of mutual entropy that is more computationally viable compared to Monte Carlo methods. We motivate the properties that such a measure is desired to have, and proceed to define, and derive, such a measure. Finally, we discuss its interpretation and visualise its advantage through simple tests cases in both the binary and multiclass classification setting.

4.3.1 Binary Classification

For binary classification, linearisation is performed on the likelihood response function.

Suppose we have trained our Gaussian process classifier using Laplace approximation with respect to a training set $\mathcal{D} = \{X, \mathbf{y}\} = \{[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T, [y_1, y_2, \ldots, y_n]\}$ with n training points. We know that the latent function $f(\mathbf{x})$ is distributed as a GP with a particular predictive mean $m(\mathbf{x})$ and covariance $k(\mathbf{x}, \mathbf{x}')$ once conditioned on the training data (4.1). From here on we omit explicitly notating the training set that was conditioned upon.

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

Let $X^\star = [\mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_{n^\star}^\star]^T$ denote the collection of $n^\star$ query points for which inference is to be performed upon. Denote $\mathbf{f}^\star$ the vector of latent function values $f_i^\star = f(\mathbf{x}_i^\star)$ at each query point. We have by definition of a GP that $\mathbf{f}^\star$ is multivariate Gaussian distributed with corresponding means $\mu_i^\star = m(\mathbf{x}_i^\star)$ and covariances $\Sigma_{ij}^\star = k(\mathbf{x}_i^\star, \mathbf{x}_j^\star)$ (4.2).

$$\mathbf{f}^\star = [f_1^\star, f_2^\star, \ldots, f_{n^\star}^\star]^T \sim \mathcal{N}(\boldsymbol{\mu}^\star, \Sigma^\star) \tag{4.2}$$

The binary prediction probability $\boldsymbol{\pi}^\star$ at the query points is obtained through passing the queried latent function random vector $\mathbf{f}^\star$ through a response function in a component wise fashion (4.3).

$$\boldsymbol{\pi}^\star = \boldsymbol{\sigma}(\mathbf{f}^\star) \quad \text{i.e.} \quad \pi_i^\star = \sigma(f_i^\star) \quad \forall i \in \{1, 2, \ldots, n^\star\} \tag{4.3}$$

As a straightforward transformation of the latent vector, the predictive probability vector $\boldsymbol{\pi}^\star$ is thus a random vector itself. The usual procedure is then to treat the expected prediction probabilities $\mathbb{E}[\boldsymbol{\pi}^\star]$ as the posterior class probabilities for further inference. However, this discards any information regarding the joint behaviour of an arbitrary collection of query points. As a result, a measure of mutual information shared amongst the query points cannot be obtained.

One straightforward approach to address this problem is to perform Monte Carlo estimation of the posterior joint distribution for class predictions via jointly sampling latent vectors from the GP, assigning class label 1 for positive latent values and -1 otherwise, and compute the Shannon entropy Shannon [1948] from the estimated joint distribution. Aside from the time complexity required for sampling enough draws for accurate joint distribution estimation, Monte Carlo estimated Joint Information Entropy (MCJIE) also has the tendency to over-represent uncertainties under small samples.

Instead, we propose using the joint distribution of the predictive probabilities $\boldsymbol{\pi}^\star$ itself as a basis of constructing a measure of mutual information. Unlike traditional approaches where inference depends only on the expectance $\mathbb{E}[\boldsymbol{\pi}^\star]$ such that structural information from the latent GP is compromised, we utilise also the covariance $\mathbb{V}[\boldsymbol{\pi}^\star]$, which contains information regarding both the latent GP and the response likelihood.

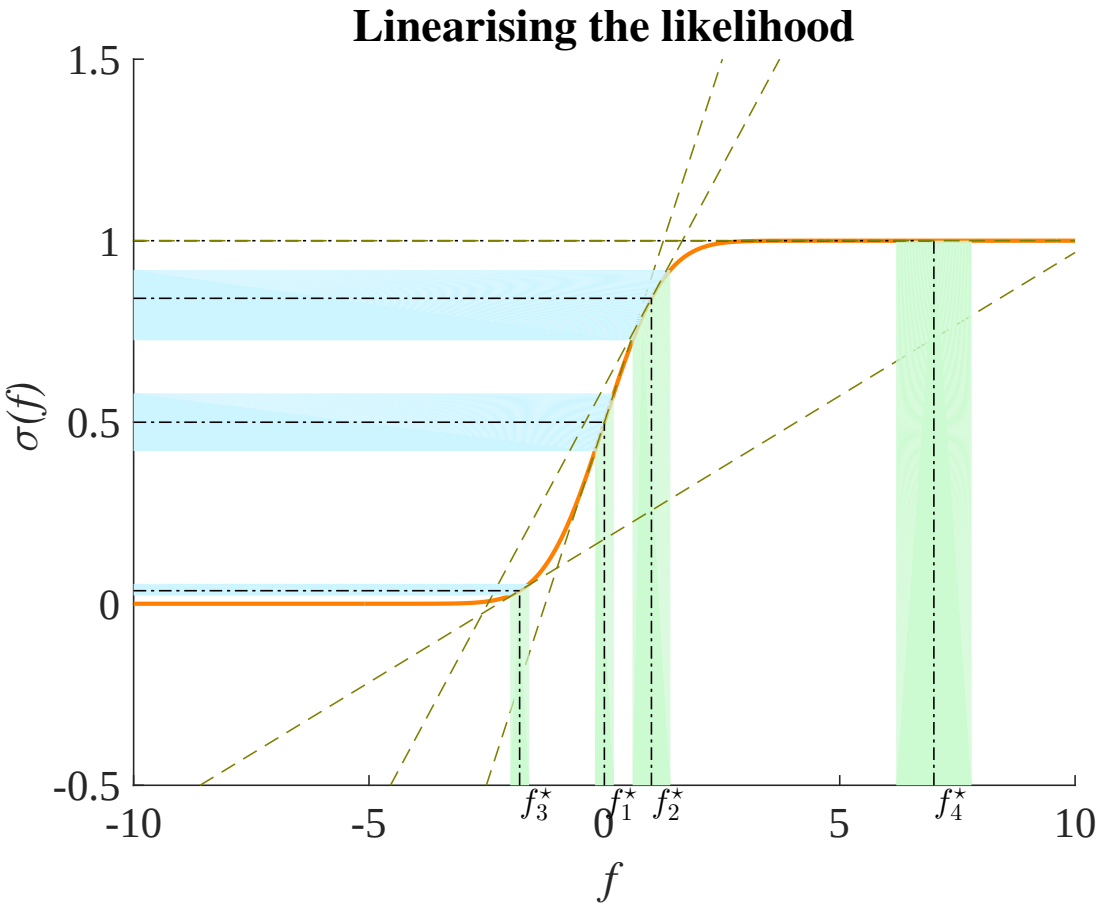


FIGURE 4.1: Linearisation accuracy for a probit response: Green shade represents the latent variance while blue shade represents the predictive variance. Gold lines show local linearisation about latent expectance.

As the predictive probabilities are nonlinear transformations of the Gaussian distributed latent vector, they are no longer Gaussian distributed. In order to retain analytical tractability, we propose linearising the response function about the latent expectance $\bar{f}_i^\star := \mathbb{E}[f_i^\star]$. Figure 4.1 illustrates the linearisation accuracy for a probit response. Observe that points with latent expectance far away from zero translate to near zero predictive variance even under high latent variance. Linearisation is thus very accurate for those points. For points with latent expectances near zero, we require the latent variances to be sufficiently small for linearisation to be accurate.

4.3.1.1 Derivation

We proceed to derive the linearisation which also serves to construct the definition of linearised differential entropy. With first order Taylor expansion, we linearise the response about the latent mean $\bar{f}_i^\star = \mathbb{E}[f_i^\star]$ (4.4).

$$\sigma(f_i^\star) \approx \sigma_L(f_i^\star) := \sigma(\bar{f}_i^\star) + \sigma'(\bar{f}_i^\star)(f_i^\star - \bar{f}_i^\star) \tag{4.4}$$

The prediction probabilities are now approximated as a linear transformation $\sigma_L(f)$ of the latent vector, so that it is also multivariate Gaussian distributed with expectance and covariance available in analytical form (4.5).

$$\begin{aligned} \boldsymbol{\sigma}_L(\mathbf{f}^\star) &\sim \mathcal{N}(\boldsymbol{\mu}_L^\star, \Sigma_L^\star) \\ (\mu_L^\star)_i &= \mathbb{E}[\sigma_L(f_i^\star)] = \sigma(\bar{f}_i^\star) \\ (\Sigma_L^\star)_{ij} &= \mathbb{C}\mathrm{ov}[\sigma_L(f_i^\star), \sigma_L(f_j^\star)] = \sigma'(\bar{f}_i^\star)\sigma'(\bar{f}_j^\star)\mathbb{C}\mathrm{ov}[f_i^\star, f_j^\star] \end{aligned} \tag{4.5}$$

We then define the linearised differential entropy $H_L^\star$ at the query points $X^\star$ to be the differential entropy for which the random vector $\boldsymbol{\sigma}_L(\mathbf{f}^\star)$ holds. Since $\boldsymbol{\sigma}_L(\mathbf{f}^\star)$ is multivariate Gaussian distributed, H_L exhibits a closed form (4.6).

$$H_L^\star := \frac{1}{2}\log\left((2\pi e)^{n^\star}\det(\Sigma_L^\star)\right) \tag{4.6}$$

4.3.2 Multiclass Classification

For multiclass classification, linearisation is performed on the softmax function σ^m which returns the corresponding predictive class probability $\boldsymbol{\pi}^m$ for class $m \in \{1, 2, \ldots, c\}$ (4.7) Rasmussen and Williams [2006]. For notational clarity we move the query star (*) to the left and use the superscript m to index the classes. The latent vector $\mathbf{f}_i := \{f_i^m\}_{m\in\{1,2,\ldots,c\}}$ represents the collection of c latent values across classes at the query point i, and is distinct from $\mathbf{f}^m := \{f_i^m\}_{i\in\{1,2,\ldots,n^\star\}}$ which represents the collection of $n^\star$ latent values across query points for class m.

$$ {}^\star\pi_i^m = \sigma^m({}^\star\mathbf{f}_i) := \frac{\exp({}^\star f_i^m)}{\sum_{l=1}^c \exp({}^\star f_i^l)} \qquad m \in \{1, 2, \ldots, c\} \tag{4.7} $$

In this derivation, we focus on the case of OVA, or one versus all, multiclass classification, where each class is trained against all other classes independently with a binary classifier. For c classes, c binary classifiers are trained independently and also performs inference independently. The normalisation is then inherently captured in the softmax (4.7). This approach avoids the Monte Carlo sampling step in the inference stage of a typical GP multiclass classifier under Laplace Approximation Rasmussen and Williams [2006], and is thus faster in computational time. Furthermore, because each classifier is trained independently, both the learning stage and the inference stage can be performed in parallel, further speeding up the process in a way that was not available under the original scheme. This is important later in under a receding horizon formulation, where the inference stage is included in the objective function of an optimiser such that repeated evaluations would benefit dramatically from shorter inference time.

4.3.2.1 Derivation

Similar to the binary case, to linearise we first find the gradient of each of the c softmax functions (4.8). For notational simplicity, the query stars (*) are omitted in this derivation except for $n^\star$.

$$ \frac{\partial \sigma^m}{\partial f_i^k}(\mathbf{f}_i) = \begin{cases} -\frac{\exp(f_i^m)\exp(f_i^k)}{\sum_{l=1}^c \exp(f_i^l)} & \text{for } k \neq m \\ -\frac{\exp(f_i^m)^2}{\left(\sum_{l=1}^c \exp(f_i^l)\right)^2} + \frac{\exp(f_i^m)}{\sum_{l=1}^c \exp(f_i^l)} & \text{for } k = m \end{cases} \tag{4.8} $$

The numerators are explicitly left in the form of products of exponentiation instead of exponentiation of sums to reflect ways to cache quantities during computation.

Hence, for each class m at query point i we compute a softmax gradient vector (4.9).

$$\frac{\partial \sigma^m}{\partial \mathbf{f}_i}(\mathbf{f}_i) := \begin{bmatrix} \frac{\partial \sigma^m}{\partial f_i^1}(\mathbf{f}_i) & \frac{\partial \sigma^m}{\partial f_i^2}(\mathbf{f}_i) & \dots & \frac{\partial \sigma^m}{\partial f_i^c}(\mathbf{f}_i) \end{bmatrix}^T \tag{4.9}$$

The linearisation is again performed on the mean latent predictions $\bar{\mathbf{f}}_i = \mathbb{E}[\mathbf{f}_i]$ so that the we can approximate the softmax $\sigma^m(\mathbf{f}_i)$ with the linearised softmax $\sigma_L^m(\mathbf{f}_i)$ (4.10) in an analogous form as (4.4).

$$\begin{aligned} \sigma^m(\mathbf{f}_i) \approx \sigma_L^m(\mathbf{f}_i) &:= \sigma^m(\bar{\mathbf{f}}_i) + \frac{\partial \sigma^m}{\partial \mathbf{f}_i}\bigg|_{\mathbf{f}_i = \bar{\mathbf{f}}_i}^T (\mathbf{f}_i - \bar{\mathbf{f}}_i) \\ &= \mathbf{c}_i^m + (\mathbf{g}_i^m)^T (\mathbf{f}_i - \bar{\mathbf{f}}_i) \end{aligned} \tag{4.10}$$

where we have notated the constants $\mathbf{c}_i^m := \sigma^m(\bar{\mathbf{f}}_i)$ and $\mathbf{g}_i^m := \frac{\partial \sigma^m}{\partial \mathbf{f}_i}(\bar{\mathbf{f}}_i)$.

To determine the distribution of the vector of softmax values across query points, we first define the following (4.11).

$$\begin{aligned} F &:= \{f_i^m\}_{m \in \{1,2,\dots,c\}, i \in \{1,2,\dots,n^\star\}} \in \mathbb{R}^{c \times n^\star} \\ \boldsymbol{\sigma}_L^m(F) &:= \begin{bmatrix} \sigma_L^m(\mathbf{f}_1) & \sigma_L^m(\mathbf{f}_2) & \dots & \sigma_L^m(\mathbf{f}_n) \end{bmatrix}^T \end{aligned} \tag{4.11}$$

We can now compute the covariance of the linearised sigmoid of a particular class m between two query points i and j, as well as the expectance at a particular query point i (4.12).

$$\begin{aligned} \boldsymbol{\sigma}_L^m(F) &\sim \mathcal{N}(\boldsymbol{\mu}_L^m, \Sigma_L^m) \\ (\mu_L^m)_i &= \mathbb{E}[\sigma_L^m(\mathbf{f}_i)] = \sigma^m(\bar{\mathbf{f}}_i) \\ (\Sigma_L^m)_{ij} &= \mathbb{C}\text{ov}[\sigma_L^m(\mathbf{f}_i), \sigma_L^m(\mathbf{f}_i)] \\ &= \mathbb{C}\text{ov}[(\mathbf{g}_i^m)^T \mathbf{f}_i, (\mathbf{g}_j^m)^T \mathbf{f}_j] \\ &= \sum_{k=1}^{c} (g_i^m)^k (g_j^m)^k \mathbb{C}\text{ov}[f_i^k, f_j^k] \end{aligned} \tag{4.12}$$

where $(g_i^m)^k$ denotes the k^{th} element of $\mathbf{g}_i^m$. The last equality arises as a result of employing OVA multiclass classification, where latent values of classes k and l, $k \neq l$, are conditionally independent given training observations.

Finally, we define the linearised entropy of the OVA multiclass Gaussian process classifier as follows (4.13).

$$H_L := \frac{1}{2} \log \left((2\pi e)^{n^\star} \det \left(\sum_{m=1}^{c} \Sigma_L^m \right) \right) \tag{4.13}$$

4.3.3 Interpretation of Model Differential Entropy

It is worthwhile to remember that linearised differential entropy is not an approximation to the usual prediction information entropy - the former is a differential entropy on a multivariate Gaussian distribution and the latter is an information entropy on the distribution of discrete class predictions combinations, which is a multivariate multinomial distribution. They have different interpretations and both can have advantageous properties under different scenarios. We propose using linearised differential entropy as an alternative acquisition function for informative path planning which can be more beneficial under specific exploration purposes.

Specifically, while the prediction information entropy (PIE) represents the confidence in the model, the linearised differential entropy (LDE) represents the the separability of classes in the feature space. In fact, LDE quantifies the ambiguity of the predictive probabilities. In particular, it is possible for GP multiclass classifiers to conclude low predictive variance even under uncertain predictive probabilities. In this case, the LDE would be high while the PIE would be low at such locations, indicating that the model is confident about its inability to separate classes Bender et al. [2013]. Similar to the bias-variance trade-off, such a situation indicate high bias, and is distinct from a model that is unconfident about its ability to separate classes, which would correspond to high variance. This demonstrates that PIE and LDE are two complementary measures of entropy that can assist each other in identifying both places of high bias and high variance.

In the case of informative path planning, a measure of mutual information is required. While the formulation of LDE readily incorporates joint distributions

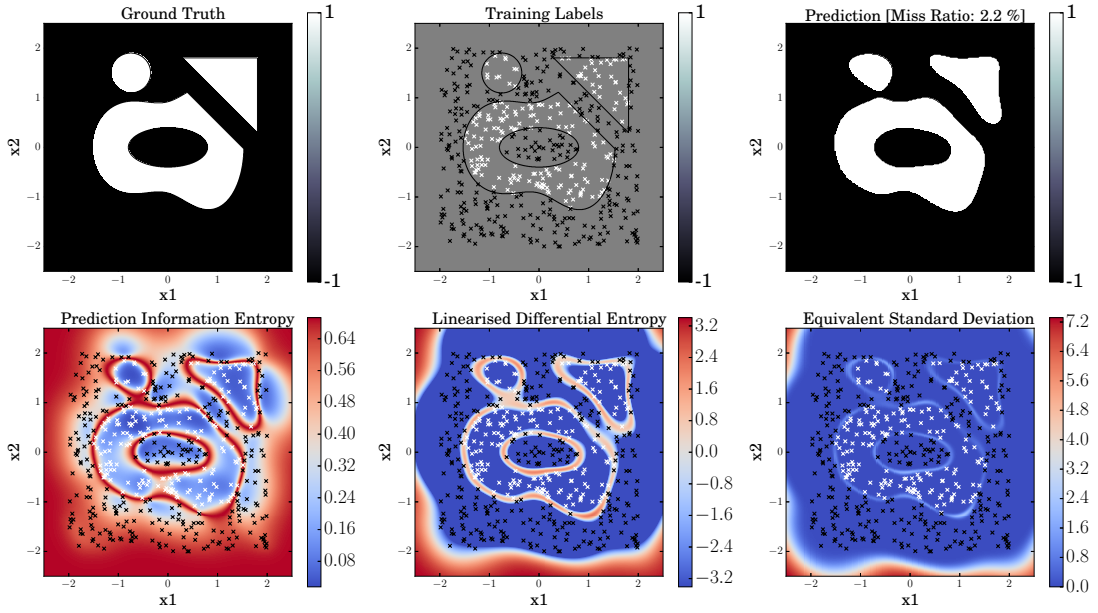


FIGURE 4.2: GP binary classifier: prediction information entropy and linearised differential entropy under abundant data

(4.13), to obtain the joint PIE one would need to perform estimations from techniques such as Monte Carlo sampling. We refer to the latter as Monte Carlo estimated Joint Information Entropy (MCJIE).

Here we compare linearised differential entropy and the usual prediction information entropy of a Gaussian process classifier. For visualisation purposes, only the marginalised entropies across the feature space are shown. We show examples where abundant information is available, so that the difference in the properties in the two measures can be emphasized.

Figure 4.2 shows a simple binary classification problem with abundant data on features x_1 and x_2, allowing a misclassification rate of 2.155%. The classifier is trained with an axis aligned Gaussian kernel with a probit response under Laplace approximation.

In this example, there are no training points around the edges shown. As a result, the GP classifier learns a slightly lower signal to noise ratio in the latent function, and bounces back to its latent GP prior for which it remains uncertain regarding class label assignments. The prediction information entropy reflects this change more rapidly, so that it is high both at the decision boundaries and wherever observations are lacking. If the acquisition function for informative exploration is

a function of the prediction information entropy, the vehicle would be suggested to explore both places with lacking observations and the decision boundaries.

On the other hand, linearised differential entropy focuses on the decision boundary as it is constructed to be only high when the latent function is near zero (figure 4.1). We can see in Figure 4.2 that the linearised differential entropy emphasizes on where the latent expectance is close to zero, and filters out the rest. If the linearised differential entropy is used as the acquisition function, the vehicle would focus on the decision boundaries within the feature space. Notice that regions far away from observations in the feature space are also assigned with high linearised differential entropy, as the latent function bounces back to its prior, so that the vehicle would explore those parts of the feature space if necessary.

This demonstrates the advantage of linearised differential entropy. We can see in Figure 4.2 that the linearised differential entropy is only high at the predicted decision boundaries. Note that the colour scale has been centred around zero differential entropy. Under such an acquisition function, the vehicle would focus strictly on the mapping the predicted decision boundaries better.

Figure 4.3 shows a similar scenario with a multi-class scenario with 4 labels. The same behaviour as the binary case is observed, where the linearised differential entropy approach pushes down the entropy level of all regions except the predicted decision boundaries.

4.4 The Receding Horizon Formulation

In this section we present a receding horizon approach to the informative path-planning problem. Specifically, we use the linearised differential entropy derived earlier as the acquisition function. We motivate the use of the approach and discuss its structure.

The receding horizon approach is inspired by the philosophy of model predictive control (MPC) in control theory, for which a continuous problem is discretised such that an optimal control problem is transcribed into a static optimisation problem at each time step. Similar to MPC, while receding horizon methods are almost always suboptimal, it provides a computationally tractable approach that is rather stable in performance. Furthermore, a receding horizon approach avoids

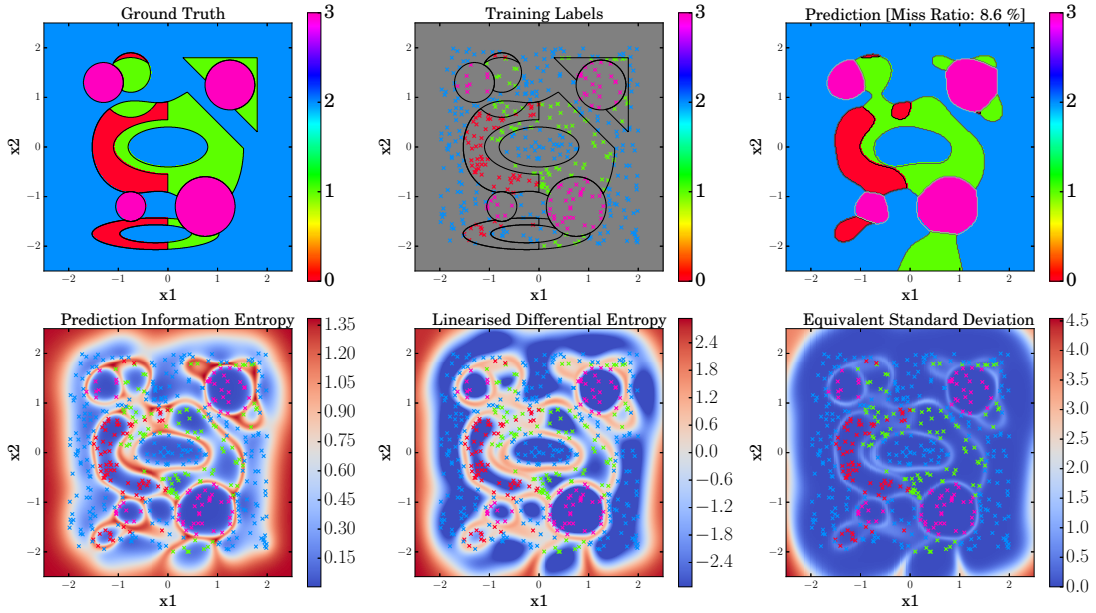


FIGURE 4.3: GP multiclass classifier: prediction information entropy and linearised differential entropy under abundant data

a myopic and greedy approach to informative path planning. Myopic approaches often result in the vehicle fixating on a region with local maximum entropy due to its inability to sacrifice immediate rewards for future rewards in a faraway region. Lastly, a receding horizon approach is simple to implement and sufficient for most missions for which strict optimality is not necessary.

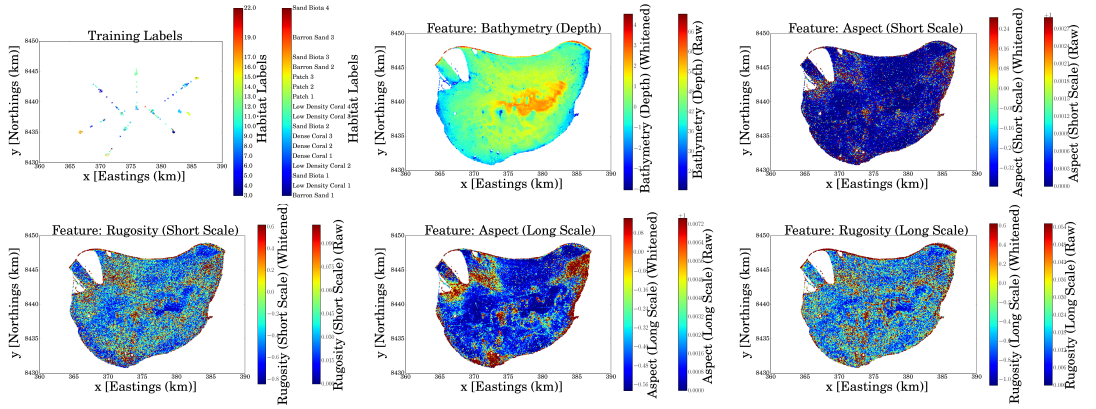


FIGURE 4.4: Scott Reef Bathymetric Features

4.4.1 Formulation and Structure

The receding horizon approach requires the selection of a horizon length and the number of control points (query points) for which the path is to be defined upon.

The horizon length plays a significant role in the performance of the method. A short horizon length tends to produce paths that are similar to a myopic approach. A horizon length that is too long, however, can be both inefficient and destabilising. For an informative path-planning scenario, looking ahead too far can have diminishing returns in its informativeness, as the vehicle's belief space would be significantly altered by the time it was supposed to follow the original proposed path.

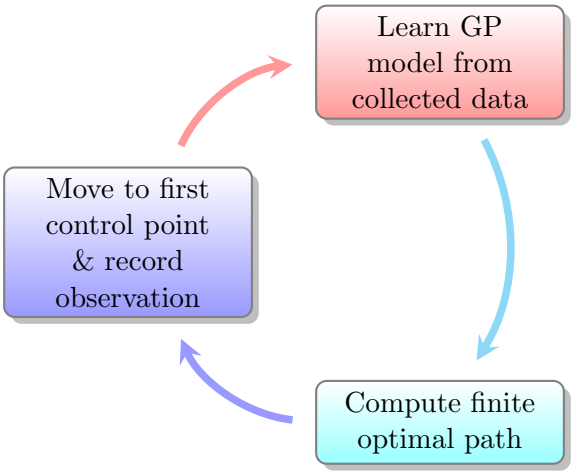


FIGURE 4.5: Basic Receding Horizon Structure

Figure 4.5 describes the basic flow of the receding horizon approach approach, which resembles the technique of model predictive control (MPC) in control theory. That is, it computes an optimal policy of finite horizon in each time step, and only executes the first action from the policy. The acquisition function to be maximised in each step is flexible, and we compare LDE acquisition (4.13) against MCJIE acquisition and other acquisition functions under the receding horizon structure in the next section. In each time step, a new path of a certain horizon length is proposed, which is discretised into a finite set of control points. The vehicle only executes the policy towards the first control point while recording observations. It then relearns the GP classifier model with new observations, and repeats the process.

Step Spacing

4.4.2 Implementation

Path Generation & Natural Coordinates

Turn Angle Limits for Smooth Paths

Feature Space Transformations

4.4.3 Computational Aspects: Optimisation Process and Bottlenecks

Feasibility and Practicality of LMDE emphasized here

4.4.4 Computational Aspects: Model Update

4.5 Informative Exploration over Scott Reef Seafloor

4.5.1 Ground Truth Generation

4.5.2 Practical Considerations

4.5.3 Entropy and Prediction Maps

4.5.4 Performance Assessment

Chapter 5

Conclusion and Future Work

Appendix A

Computational Aspects of Gaussian Processes

A.1 Numerical Stability

A.1.1 Cholesky Decomposition

A.1.2 Solving Triangular Matrix Equations

A.1.3 Cholesky Jittering

A.2 Time Complexity

Reducing computational time

A.2.1 Numpy and Vectorisation

A.2.2 Cholesky Update and Downdate

A.2.3 Caching learned GPs for fast prediction

A.2.4 Parallelisation of GP learning and hyper-parameter batching

A.2.5 Parallelisation of GP prediction and relevant subtleties

A.2.6 Fast vectorised GP drawing for regression and classification

A.3 Spatial Complexity

Reducing memory requirements

A.3.1 Symmetry of AVA multiclass classifier

A.3.2 Creating predictor objects to modularise prediction

A.4 Time & Spatial Complexity

A.4.1 Avoiding full covariance computations

A.4.2 Taking advantage of diagonal log-likelihood Hessians

Appendix B

Other Approximations for Gaussian Process Classification

B.1 Expectation Propagation

B.2 Variational Inference

Bibliography

Bender, A., Williams, S. B., and Pizarro, O. (2013). Autonomous methods for environmental modeling and exploration. In *Proceedings of the Robotic Science and Systems*, pages 1–8.

Brooks, A., Makarenko, A., Williams, S., and Durrant-Whyte, H. (2006). Parametric POMDPs for planning in continuous state spaces. *Robotics and Autonomous Systems*, 54(11):887 – 897. Planning Under Uncertainty in Robotics.

Colbo, K., Ross, T., Brown, C., and Weber, T. (2014). A review of oceanographic applications of water column data from multibeam echosounders. *Estuarine, Coastal and Shelf Science*, 145:41 – 56.

Friedman, A., Pizarro, O., Williams, S. B., and Johnson-Roberson, M. (2012). Multi-scale measures of rugosity, slope and aspect from benthic stereo image reconstructions.

Friedman, A., Williams, S., and Toohey, L. (2015). *Squidle*.

Holoborodko, P. (2010). Central differences.

IMOS (2009). *Scott Reef Lagoon - WA*. Integrated Marine Observation System.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2010). Gaussian processes for object categorization. *International Journal of Computer Vision*, 88(2):169–188.

Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284.

Lang, T., Plagemann, C., and Burgard, W. (2007). Adaptive non-stationary kernel regression for terrain modelling. In *In Proc. of the Robotics: Science and Systems Conference (RSS*.

Marchant, R., Ramos, F., and Sanner, S. (2014). Sequential bayesian optimisation for spatial-temporal monitoring. *Conference on Uncertainty in Artificial Intelligence*.

Niedzielski, T., Åge Høines, Shields, M. A., Linley, T. D., and Priede, I. G. (2013). A multi-scale investigation into seafloor topography of the northern mid-atlantic ridge based on geographic information system analysis. *Deep Sea Research Part II: Topical Studies in Oceanography*, 98, Part B:231 – 243. ECOMAR: Ecosystems of the Mid-Atlantic Ridge at the Sub-Polar Front and Charlie-Gibbs Fracture Zone.

NOAA (2014). *How much of the ocean have we explored?* National Oceanic And Atmospheric Administration, United States Department of Commerce.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Process for Machine Learning*. The MIT Press.

Rigby, P., Pizarro, O., and Williams, S. B. (2010). Toward adaptive benthic habitat mapping using gaussian process classification. *Journal of Field Robotics*, 27(6):741–758.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.

Steinberg, D. M., Pizarro, O., and Williams, S. B. (2015). Hierarchical bayesian models for unsupervised scene understanding. *Computer Vision and Image Understanding*, 131:128 – 144. Special section: Large Scale Data-Driven Evaluation in Computer Vision.

Tong, C. H. (2013). Research - 3D SLAM for Mapping Planetary Worksite Environments. Accessed: 15/04/2015.