Anomaly Detection in Vessel Track Data

Mark Smith

Department of Engineering

University of Oxford

A thesis submitted for the degree of

Master of Science (Research)

Michaelmas 2013

Acknowledgements

This work was funded by ISSG, Babcock Marine & Technology Division, Devonport Royal Dockyard. Personally, I would like to thank: Anthony Chiswell, Stuart Cox and Philip Burns-O'Brien for their support, motivation and mentorship of which I am deeply indebted; Iead Rezek, Steven Reece and Stephen Roberts for their endless patience and guidance, without which none of this work would have been possible; all the members of the Machine Learning Research Group (MLRG) who helped and supported me.

My thesis contains work from several joint-authored peer-reviewed publications, which are listed below.

M. Smith, S. Reece, S. Roberts and I. Rezek. Online Maritime Abnormality Detection using Gaussian Processes and Extreme Value Theory. In *Proceedings of IEEE 12th International Conference on Data Mining (ICDM), Brussels, Belgium*, pages 645-654, 2012.

M. Smith, S. Reece, S. Roberts, I. Psorakis and I. Rezek. Maritime Abnormality Detection using Gaussian Processes. *Knowledge and Information Systems*, 45(3):717-741, 2013.

Abstract

This thesis introduces novelty detection techniques that use a combination of Gaussian processes, extreme value theory and divergence measurement to identify anomalous behaviour in both streaming and batch marine data. The work is set in context by a review of current methodologies, identifying the limitations of current modelling processes within this domain.

Marine data modelling is first improved by endowing the Gaussian process with the capacity to model both first order and second order dynamics; enhancing maritime data modelling through exploration of appropriate Gaussian process kernels. Gaussian processes are then used to forecast probable future vessel positions. The concept of combining the predictive uncertainty from the Gaussian process with extreme value distributions is then introduced. This provides a means of detecting anomalous vessel dynamics given the previously learnt model. The process is made amenable to online operation through adaption of the Gaussian process to sequential updates. The latter allows the model to be updated in an efficient online manner, after confirming that received data lies within the probability bounds of the model forecast behaviour. Finally a means of measuring distance between functions is introduced, which is used to identify communities of similar vessel types based on the underlying vessel dynamics. This is used to address the issue of vessel class (i.e. fishing vessel, cargo vessel etc.) misrepresentation through detection of anomalies between the inferred vessel class and the class broadcast by the vessel.

Contents

Chapter 1

Introduction

The global picture of maritime traffic is large and complex, consisting of dense volumes of (mostly legal) ship traffic. Techniques that identify illegal traffic could help to reduce the impact from smuggling, terrorism, illegal fishing etc. In the past, surveillance of such traffic has suffered due to a lack of data. However since the advent of electronic tracking the amount of available data has grown beyond an analyst's ability to process without some form of automation. One part of the analyst's workload lies in the detection of anomalous behaviour in otherwise normal appearing tracks. The primary concern of this thesis is to investigate automated methods of anomaly detection within vessel track data. This is achieved through the exploitation of techniques from the areas of machine learning and anomaly detection.

A variety of methods have previously been applied to the detection of anomalies and these are briefly reviewed within Chapter 2. The review highlights that common between all methods are two main tasks; creating a model of normality (free from the presence of anomalies), and creation of a metric from this model which (allowing for some quantifiable variability) identifies a point as anomalous. These tasks are inherent within the two main uses for the detection of anomalies; *accommodation* and *discordancy*. Accommodation is the task in which the goal is create a model of normality that does not include anomalous observations, whereas discordancy tests provide a metric indicator of a point being an anomaly. Both have different aims but within each is some model of normality and a measure of deviation. Chapter 2 concludes with a comparative review of methods currently being

applied within the maritime domain. It is noted that the field of marine anomaly detection has employed a variety of methods including neural networks Rhodes et al. (2005), Bayesian networks Mascaro et al. (2011), support vector machines Li et al. (2006), Gaussian processes Will et al. (2011) and Kalman filters Laws et al. (2011). Assessment of the performance of these different methods in the marine domain is a difficult task as there exist no established benchmarks of what are considered as marine anomalies, therefore hindering comparison Laxhammar (2008). However the review highlights that current methods within this domain fail to account (in a principled manner) for outlying observations when detecting anomalies.

Given that there exist many possible types and interpretations of marine anomalies, the scope of this thesis begins by considering only the identification of anomalies based on previously observed dynamics of a given vessel track i.e. to identify whether subsequent dynamics are unusual conditioned on previous non-anomalous observations. Many of the modelling techniques typically applied to identify marine anomalies are based on Bayesian approaches, which allow the incorporation of our prior belief into the underlying system model. Chapter 3 provides a brief review of the Bayesian methodology and provides derivations under this framework of the two methods employed in this thesis, namely; Gaussian processes and the Kalman filter/smoother. The focus is on the use of Bayesian methodologies, in order to allow for the inclusion of common-sense knowledge and observational evidence into the model. The methodologies chosen are derived assuming a Gaussian distribution over some marginal subset, previous work having shown that such a distributional assumption can be used to capture the normal characteristics of marine data Laxhammar (2008); Laxhammar et al. (2009). The relationship and interconnections between the two methods are discussed, and the benefits (in this instance) of a non-parametric Bayesian approach to regression (Gaussian processes) highlighted.

Gaussian processes are made amenable to vessel track modelling in Chapter 4. The modelling process is improved through the incorporation of vessel dynamics into the Gaussian process model, using techniques originally developed for the Kalman filter. Specifically, differential equations describing an object with a near constant velocity and acceleration are expressed in kernel form, for application

within the Gaussian process.

The issue of identifying anomalous dynamics is addressed in Chapter 5. Extreme value statistics are combined with Gaussian processes, enabling the variability of the population of normal data to be accurately captured and hence anomalous movements identified. Sequential updates of the Gaussian process probability distribution are made governing these extreme values, enabling context sensitive decisions. The patterns achieved by means of linking this distribution to a sequential Gaussian process model which regresses the vessel's track and forecasts a distribution over future data. Gaussian process extreme value theory is then applied to a one-dimensional feature space, allowing sudden changes in vessel dynamics to be detected.

In Chapter 6, the focus is moved toward identifying whether a vessel track in its entirety can be considered anomalous. This is achieved by clustering vessels (modelling communities) based on the underlying dynamics mandated by a vessel's class (i.e. fishing vessel, cargo vessel etc.). Modelling of the data is achieved through application of a multiple-output Gaussian process to a vessel's latitudinal and longitudinal data, ensuring vessel dynamics are suitably captured. The Hellinger distance between the inferred Gaussian process models is then used as a measure between vessel tracks, which can also be interpreted as a distance between functions. Bayesian non-negative matrix factorization is then used as an unsupervised means of identifying community structure, capturing communities of vessels with similar dynamics. Anomalies can be identified by noting those tracks not assigned to a community. Additionally, subsequent anomalous vessel tracks can be detected by noting discrepancies between the vessel class broadcast by the vessel (as mandated in Automatic Identification System data) and the labelled community to which the track is assigned.

Finally the Conclusion highlights possible future work and improvements which can be made to both the modelling process and anomaly detection schemes. The conclusion additionally provides some concluding remarks on the limitations of the current work.

Chapter 2

Anomaly Detection Review

Anomaly detection is a widely researched problem in many fields, including those of statistics and machine learning. Although anomalies are often considered to be indicative of an error or noise their detection can also provide enhanced system information, indicating a change from normal operation. Techniques used to identify such changes have wide applicability to a number of areas including:

- Intrusion detection – Detecting unauthorised access in computer networks via detection of anomalous network behaviour/traffic Javitz and Valdes (1991).
- Activity monitoring/fraud detection – Detecting trading fraud by monitoring suspicious trading activities or banking fraud detection via detection of anomalous purchases Viaene et al. (2002).
- Fault diagnosis – Detection of faulty goods/systems via detection of anomalous sensor behaviour/response Petković et al. (2012).
- Image analysis – Identifying novel or misclassified features in images Yonga et al. (2012).
- Marine anomalies – Identifying illegal shipping activity Zandipour et al. (2008).

The emphasis of this chapter is to survey briefly anomaly detection techniques, providing the reader with a structure within which to consider future work. A

review of current anomaly detection work within the marine domain will then be undertaken. In the process some seminal papers highlighting traditional methods and newer approaches will be reviewed to provide background theory. It should be noted that the field of anomaly detection is incredibly broad, and this chapter does not attempt to be inclusive. For detailed reviews the reader is directed to Markou and Singh (2003a,b); Chandola et al. (2009).

2.1 The Anomaly Hypothesis

An anomaly has many different interpretations depending on the context in which it is used, for example it may refer to a data point arising from a different distribution, measurement error, population variability or execution error. However, fundamentally an anomaly is a data point which stands out in contrast to the other data points around it Grubbs (1969). It is the task of anomaly detection to detect whether this data point deviates sufficiently from the inherent variability of the population, so as to quantify the likelihood of the data point as having arisen due to some other mechanism. An effective technique should therefore be capable of recognising and modelling data points that occur due to extreme events (outliers), as well as events which may appear to be anomalous, yet actually arise from the same underlying distribution Dixon (1950).

However, the prior definition of an anomaly would appear to become slightly confused when authors talk of a model of anomalies, see for example Münz et al. (2007). If the data generating mechanism, in addition to the anomaly generating mechanisms is captured, then the definition of an anomaly as a data point contrasting to the other data points around it would appear inadequate. No data points would appear to stand out under the model. The distinction is therefore usually made that an anomaly is a data point that stands out in relation to the anomaly free model describing the data generating mechanism under investigation.

Anomalies can be handled in one of three ways; accommodation, identification or rejection Barnett (1978). In all of these approaches the presence of the anomaly needs to be explained by a suitable model. Many models exist, and selection of an appropriate model depends on how it is hypothesised the presence of anomalies

can be explained. A discussion of the common hypotheses regarding the anomaly generating mechanism is important in order to aid understanding of how anomalies can be handled, as described in the following sections.

All anomaly handling approaches begin with an initial null hypothesis H that the data generating mechanism is described by a distribution, F. If a point x_i (where i is the index of the spurious data point) in a sample of n data points is identified as anomalous then an alternative hypothesis $\bar{H}$ can be proposed that explains the presence of the anomaly. A number of common hypotheses are typically considered to explain the presence of anomalies, these are known as the deterministic alternative, inherent alternative, mixture alternative and slippage alternative hypotheses Barnett (1978). The first of these hypotheses, the deterministic alternative, states that,

$$H : x_j \in F\,(j = 1 \ldots n)\,, \qquad \bar{H} : x_j \in F\,(j \neq i)\,. \tag{2.1}$$

The hypothesis proposes that data points are tested against the distribution describing the data generating mechanism. No attempt is made to model the anomaly generating mechanism. Such an approach is useful if a distribution can be readily inferred that describes the distribution on F. A threshold probability can then be used to identify likely anomalous data points. An extension to this approach, the inherent alternative hypothesis rejects the entire distribution if a sample is found to be anomalous,

$$H : x_j \in F\,(j = 1 \ldots n)\,, \qquad \bar{H} : x_j \in G \neq F\,(j = 1 \ldots n)\,. \tag{2.2}$$

In this formulation the rejection of the sample triggers rejection of the null hypothesis for the sample in favour of the alternative hypothesis, that the whole sample arises from some distribution G. The mixture alternative hypothesis assumes anomalies reflect the small chance λ that observations arise from some distribution G, quite different from the initial distribution F,

$$H : x_j \in F\,(j = 1 \ldots n)\,, \qquad \bar{H} : x_j \in (1 - \lambda)\,F + \lambda G\,(j = 1 \ldots n)\,. \tag{2.3}$$

The mixture hypothesis assumes that all the distributions are appropriately scaled by a factor λ, termed the mixture factor. The distribution G therefore needs to have appropriate form e.g. greater dispersion than F. Alternatively the distribution of anomalies can be assumed to arise independently from the initial model F, as in the slippage alternative hypothesis,

$$H : x_j \in F\,(j = 1 \ldots n)\,, \qquad \bar{H} = x_j \in F\,(j \neq i) \text{ and } x_i \in G. \tag{2.4}$$

Examples involving slippage of location, or of dispersion, respectively, arise where F has mean μ and variance σ^2, and G has the same form as F but mean $\mu + a$ $(a > 0)$ or variance $b\sigma^2$, $(b > 1)$. As previously noted, anomalies can be handled in one of three ways; accommodated, identified or rejected. Accommodation methods provide a means whereby the distribution of the underlying data generating mechanism can be inferred free from the effects of anomalies. Accommodation methods therefore seek to explain the presence of anomalies in some manner, such as via the mixture and slippage alternative hypotheses. Methods of identifying anomalies (also known as discordancy testing) provide a means of testing whether a data point, or points, are significantly different from the inherent variability of the reference distribution, as per the deterministic alternative hypothesis. Rejection methods involve rejecting the current model of the data generating mechanism, in favour of a model explaining the presence of the detected anomaly as per the inherent alternative hypothesis.

All anomaly handling approaches are inherently interlinked. To reject a model data points must be tested against the model, requiring identification. To identify anomalous data points an anomaly free reference distribution must be produced, requiring accommodation. To accommodate anomalies and produce an anomaly free reference distribution suitable models describing the anomaly generating mechanism and anomaly free reference distribution must be discovered, requiring rejection.

2.2 Discordancy Tests

Identification of anomalies involves testing a data point to determine whether the tested value is significantly different from the underlying reference distribution, as per the deterministic alternative model. Reference values are used to define regions where the test statistic is unlikely to lie, which if exceeded disprove the null hypothesis. The major categories of anomaly discordancy, as categorised by Barnett (1994), can be used to provide a means of analysing anomaly detection techniques.

Excess/Spread Statistics Metrics in which the discordancy measure consists of a ratio of the difference between a potential anomaly and its nearest or next-nearest neighbour to the range are referred to as excess/spread statistics. Anomalies are detected if the ratio is in excess of a given reference value. Alternatively some other measure of spread of the sample can be used (possibly omitting the anomaly and other extreme observations), as illustrated in Equation 2.5 where z_n is the ordered set of x_n data points within a sample of n points. If $Q > Q_{ref}$, where Q_{ref} is a reference value corresponding to the sample size and confidence level, then the questionable point is rejected.

$$Q = \frac{z_n - z_{n-1}}{z_n - z_1}. \tag{2.5}$$

In univariate data a commonly used test is the Dixon test, in which it is assumed that the underlying model is Gaussian Irwin (1925); Dixon (1950). However, such tests can prove inaccurate if multiple anomalies are present within the data. The presence of several anomalies skews the test statistic, an effect known as masking. The spread statistic approach to anomaly detection has also been adapted to high dimensional data Latecki et al. (2007). In Latecki et al. (2007) a kernel density estimator was applied to each of the given data points, providing an estimate to the data density. This value was then scaled by the average density of its m nearest neighbours and the resulting value used to identify anomalies. The higher the ratio the more likely a point is to be an anomaly. However, it has been observed that techniques such as these become less useful when used to detect anomalies in sparse higher dimensional data Aggarwal and Yu (2001).

This implies that such methods would fail to account for extremal values within sample data, such values typically being sparse.

Range/Spread Statistics These metrics replace the numerator with the sample range, as in Equation 2.6. Again, if $Q > Q_{ref}$, where Q_{ref} is a reference value corresponding to the sample size and confidence level, then the questionable sample is rejected. The denominator may also be replaced by a restricted sample analogue, independent estimate, or known value of a measure of spread of the population, s Barnett (1994).

$$Q = \frac{z_n - z_1}{s}. \tag{2.6}$$

Anomalies identified by such statistics do not indicate whether the anomaly is an upper or lower anomaly, only that the sample is anomalous.

Deviation/Spread Statistics The issue of failing to indicate whether a point is an upper or lower anomaly, as in the previous test, can be partly offset by setting the numerator as a measure of the distance of an anomaly from some measure of central tendency in the data, as in Equation 2.7. Anomalies can be identified if $Q > Q_{ref}$, where Q_{ref} is some measure of spread or deviation of the data.

$$Q = \frac{z_n - \bar{z}}{s}. \tag{2.7}$$

In applications where the underlying distribution can plausibly be treated as Gaussian, it is commonplace for anomalies to be identified via a number of standard deviations from the mean, for example points further than 2σ Imai et al. (1995). However, as highlighted in Verma (1997), this method does not scale well with the number of observed data points; the expected proportion of entries classified as anomalies only converges to the desired percentile as the number of observations increases indefinitely Basmann (2003).
Spread statistics have been applied to a variety of machine learning techniques to detect anomalies, including neural networks Bishop (1993) and graph based methods Pincombe (2005). In Bishop (1993) training data was used to estimate

a reference distribution, points exceeding a fixed number of standard deviations from the reference distribution were deemed anomalous and not passed to the neural network. In Pincombe (2005) graph based features such as diameter distance, edit distance, and maximum common subgraph vertex distance were extracted and modelled using an ARMA processes. Points were identified as anomalous if they exceeded 2σ from the ARMA mean.

In multivariate space, methods for measuring the distance between a measure of central tendency and a new sample have included the Euclidean distance, however this ignores the covariance structure and treats all distances equally. A classical way of giving weight to highly correlated variables and less weight to those with large variances is to use the Mahalanobis distance Franklin et al. (2000). Reference values can then be used to determine whether a sample is an anomaly with respect to the expected true distribution, however these reference values only allow for the testing of typically one or two values Fung (1988); Penny (1996). Additionally the Mahalanobis distance function uses the inverse covariance matrix to reflect statistical correlations between function features. Inverting this matrix is computationally demanding for feature vectors with a large number of dimensions and becomes prohibitive if adaptive sequential analysis is required.

Although the Euclidean distance does not take into account the relationship that may exist between samples, a weighted Euclidean distance has previously been applied as a discordancy metric to detect anomalies in multivariate data Münz et al. (2007). In Münz et al. (2007) the distance was empirically normalised by the range of the feature distance being measured. Normalising the distance in this manner offers the advantage of computational speed over the Mahalanobis distance. If the measured distance was greater than some predefined reference value the new sample was treated as anomalous.

The Euclidean distance has also been used to identify novel data from neural networks Augusteijn and Folkert (2002). In Augusteijn and Folkert (2002) the distance between the output pattern from the network and the output patterns for each one of the target patterns (the patterns used as targets during training) were calculated. If the distance between patterns was above a given reference value then the input pattern was considered to belong to a novel category.

The effectiveness of deviation/spread methodologies toward anomaly detection is

reliant on the selection of an appropriate reference value, above which a point can be classified as anomalous. In the absence of sufficient data this value is typically achieved through ad hoc procedures Laxhammar (2011). Values derived in such a manner often fail to account for extremal observations, as such observations are by their nature infrequent and hence data is limited or absent. Additionally such reference values once learnt on training data are typically static Laxhammar (2011), thus they fail to adapt as additional data is received.

Sums of Squares Statistics Test statistics expressed as ratios of sums of squares for the restricted (restricted sample refers to the calculation based upon the total sample minus outliers) and total samples Barnett (1994), have also been applied to test for anomalies. Such methods detect anomalies through noting a significant difference in the residuals when the test point is included. A reference value is then set, above which a significant change in the residual value indicates an anomaly Ahmed et al. (2007), thus such methods are also dependent on the appropriate selection of reference value.

Extreme/Location Statistics Another class of test statistics used to test for discordancy are those which take the form of ratios of extreme values to measures of location, as in Equation 2.8. Anomalies can be identified if $Q > Q_{ref}$, where Q_{ref} is some measure of spread or deviation of the data.

$$Q = \frac{z_n}{\bar{z}}. \tag{2.8}$$

Of particular note within the field of extreme/location statistics is extreme value theory (EVT) which is discussed in more detail later in Chapter 5 of this thesis. Extreme value theory is a branch of statistics which deals with modelling the extreme values within a distribution. When modelling the maxima of a random variable, extreme value theory plays the same fundamental role as the central limit theorem plays when modelling sums of random variables. In both cases, the theory informs us regarding the limiting distributions. These limit distributions provide a quantifiable means of differentiating between anomalies and extreme values and have previously been used to provide an adaptable bound, for identification of anomalies Roberts (2000); Clifton et al. (2011). In Roberts (2000);

Clifton et al. (2011) the parameters of the extreme value model were estimated based on the density of data points, allowing the reference value to adapt to the number of sample observations. It should be noted that, in the case of the reference distribution being unknown, it must be decided how much prior information is included in the selection of the limiting form of the prior model Burridge and Taylor (2006).

Some models, in particular Gaussian distributions, describe the data density very well near the mean, but poorly in the tails. We expect this property in regions of locally high data density, due to the central limit theorem. However the latter assumes independent identically distributed data, which is rarely found in practice. Even when the assumptions of the central limit do apply, it only guarantees that the absolute error in the normal approximation goes to zero. However, the theorem does not guarantee that the relative error approaches zero. Therefore in the extremes (e.g six standard deviations out) the absolute error may be small, but the relative error in modelling extreme values is large Nolan (1999); DuMouchel (1983).

Higher-Order Moments Third and fourth order moments namely, skewness and kurtosis, although not specifically designed for assessing anomalies, have previously been applied to anomaly detection Gu et al. (2008). In Ren and Chang (2008) the data was assumed to follow a Gaussian distribution, of which the corresponding skewness and excess kurtosis is equal to zero. Anomalies were detected if the absolute values of the skewness and kurtosis exceeded an appropriate reference value. Although this is a reasonable procedure, it is susceptible to masking when neighbouring anomalies are present. Furthermore, anomalous observations detected using such methods may arise due to perfectly valid outlying values causing moments to indicate an anomaly Welling (2005).

Wilcoxon Statistics (W-Statistics) The W-statistic is an estimate of how closely a set of observations matches a Gaussian distribution, providing a statistic that is the ratio of the square of a linear combination of all the ordered sample values to the sum of squares of the individual deviations about the sample mean Shapiro and Wilk (1965). Previously it has been used in anomaly detection by

comparison of the W-statistic obtained for a set of observations to a given reference value Collins (2008), and where samples exceeding the reference value were identified as anomalies. However, such methods also fail to account for outlying observations, which are not accurately modelled by the Gaussian distribution and therefore skew the W-statistic.

2.3 Accommodation of Anomalies

Techniques which provide a means of inferring the underlying population free from the effects of anomalies within the random sample are known as accommodation methods, also referred to as robust methods. The major categories of anomaly accommodation, as categorised by Barnett (1994), can be used as a further means of analysing anomaly detection techniques.

Linear Order Statistical Estimators (L-estimators) L-estimators take the form of a linear combination of weighted ordered values, as in Equation 2.9. The weighted ordered values being lower in the extremes than in the remainder of the data.

$$\text{L-estimator} = \sum_{i=1}^{n} c_i z_i, \tag{2.9}$$

where $c_i \in \mathbb{R}$. These methods include techniques such as winsorizing and trimming (in which the extremes are effectively weighted zero). These concepts have been extended to the multivariate domain, in order to better estimate the true measure of central tendency Chaudhuri (1996). However the problem still exists in setting the extent of the trimmed region.

Minimum Estimators (M-estimators) M-estimators are a class of estimators which are obtained as the minima of sums of functions of the data, Equation 2.10. Consider a function $f(x_1 \dots x_n, \theta)$, where θ is the functions' parameter space. M-estimators are solutions of θ which attempt to minimise the residual

between this function and the data.

$$\text{M-estimator} = \overset{\text{Min}}{\theta} \left(\sum_{i=1}^{n} f\left(x_i, \theta\right) \right). \tag{2.10}$$

This approach can be used to accommodate anomalies by considering the data as being generated from two or more distinct models. For example, a model describing normal operation, and one describing the generation of anomalies. Using the M-estimator approach the effect of different parameters and functions can be considered, the residual error can then be used to discover the models and parameters which best fit the available data Lauer (2001). Different types of M-estimators can be considered to achieve this. In Lauer (2001) expectation maximisation and data augmentation were investigated to fit a Gaussian mixture model. In Scott (2004) the minimum distance method was used to fit mixture models to multivariate data, creating a robust form of density estimator.

Rank Estimators (R-estimators) The ranks $R_1 \ldots R_n$ of observations $x_1 \ldots x_n$ are invariant under all classes of monotone transformations. This invariance property yields robustness of rank-based tests against anomalies and other distributional departures, and hence, estimators of location based on rank tests are expected to enjoy similarly robust properties. Such properties have previously been exploited in Huang (2013), and anomalies identified based on a data point's ranked mutual closeness relative to observations over a number of samples. However, one immediately obvious disadvantage of such rank based testing is that the magnitude of observations is not considered. Thus outlying observations may be ranked last relative to a reference set, resulting in their incorrect identification as an anomaly.

2.4 Detection of Marine Anomalies

Research related to automated anomaly detection in the marine domain has recently attracted increased attention. Regulations regarding the mandatory use of Automatic Identification System (AIS) transponders for vessels of 300 gross tonnage have resulted in a significant increase in vessel tracking data. The previous

sections identified key techniques used in anomaly detection, and these are now used to aid analysis of methods currently being applied to the marine domain.

Many formats and sources of marine traffic data exist, including AIS and High Frequency (HF) RADAR data, both of which have been shown to be effective for providing maritime situational awareness Vesecky et al. (2009). Various techniques have been applied to the identification of anomalies within the data including: neural networks, Bayesian networks, support vector machines and Kalman filters. Assessing the performance of the different methods is a difficult task as there exist no established benchmarks of what are conceived as marine anomalies by experts, therefore hindering comparison Laxhammar (2008).

Marine anomalies can be considered in a variety of contexts, which can lead to different anomalies being identified for the same input data. For example anomalous movement can be considered at different time scales, comparing a vessel track to its historical routes may identify an anomalous movement with respect to its history. Alternatively the incoming data could be analysed with respect to the local time series, i.e. the movement is considered anomalous through consideration of the vessel's previous location and the underlying dynamics of the vessel; a sudden change may be indicative of evasive manoeuvring. The time series model has the advantage that it can be used in online analysis, but it may miss patterns at a broader scale Mascaro et al. (2011). Other indications of anomalous behaviour within the data are: deviations from a standard route, unexpected AIS activity, unexpected port arrival, close approach and zone entry Lane et al. (2010). What may be considered as normal behaviour may also vary between contexts e.g. class of vessel, time of day and tidal status. The identification of marine anomalies therefore depends primarily on the data available and the types of marine anomalies which are to be identified.

The standard approach to marine anomaly detection fits a Gaussian mixture model to maritime spatial data, in order to detect temporally-anomalous deviations. In Laxhammar (2008) data points were assigned to the component function with the highest likelihood of generating the data. This followed the inherent alternative hypothesis and used an M-estimator (a greedy form of Expectation Maximisation) to fit an appropriate mixture model to the data. In Laxhammar et al. (2009) this technique was compared to another popular method of

anomaly detection, the Kernel Density Estimator (KDE), for application to maritime data. Both methods were used to generate a probability density function of assumed normality. Once a suitable model was discovered the authors followed a deterministic hypothesis, rejecting subsequent observations as anomalous if the likelihood of the track belonging to the model was beyond a set reference value. It was demonstrated that KDE (an L-estimator) was able to more accurately characterise features in the data, such as sea lanes. This was attributed to the appropriate selection of kernel width from the data, allowing such features to be characterised. By comparison, when a Gaussian mixture model was applied to the same training data, the model failed to identify the sea lanes within the data set. However, results from the anomaly detection experiment showed no significant difference between the two methods. Moreover the results for both models were quite disappointing when tested against simulated anomalous vessel tracks. The poor results were attributed to poor feature selection and partitioning of the data, highlighting the importance of appropriate choice of both feature data and data partitioning. It was also noted in Laxhammar (2008) that one of the main challenges in configuring a system for anomaly detection is in setting an appropriate reference value for anomaly detection.

In Rhodes et al. (2005) a fuzzy ARTMAP neural network was used to identify marine anomalies. The neural network was used to cluster the data, associating the identified clusters with known anomalous and non-anomalous vessel motion patterns (e.g. speed, course and position) for various geographic regions, as labelled by a domain expert. This assumed a slippage alternative hypothesis in which the distribution of anomalies was assumed to rise independently from the model of normality. Labelled test data was used to form clusters by comparing the labelled input feature pattern to patterns assigned to a cluster. A single parameter known as the vigilence setting determined the level of generality or specificity between the input feature and the cluster. If the match between patterns was acceptable the input pattern was incorporated into the cluster, otherwise the algorithm raised the level of vigilence to correctly learn the feature pattern for the associated pattern label. Once the model was learnt the authors assumed a deterministic hypothesis and identified observations as anomalous if they fell outside a set reference value measured using a city block distance deviation statistic

from a cluster centroid. However, it was noted in Bomberger et al. (2006) that such analysis only identified specific types of marine anomaly. For instance, it was noted that two events considered in isolation may be considered normal, but their occurrence in a specific order or at a particular temporal interval may warrant closer examination on the part of an operator/analyst. In order to address such issues the work was extended in Bomberger et al. (2006) to detect temporal anomalous deviations. This required a mechanism whereby future vessel locations could be predicted. To achieve this Bomberger et al. (2006) divided the map area into grid cells and used the fuzzy ARTMAP network to learn a set of weights between cells, the weights corresponding to the typical route taken by a vessel. Each time the vessel was observed the weighted distribution of movements for the vessel was updated along its track, the update also being scaled by the prior weights. In this manner a model of normality was formed for typical vessel tracks, deviations from which could be identified as anomalous, as per the deterministic hypothesis. This offered the advantage that if during the learning phase an anomaly was encountered, then the probability of the vessel following this track would become lessened as more normal data was observed. However, the prediction was shown to be weak in open water, the number of possible routes increasing and the probability distribution becoming more evenly spread out. This was subsequently addressed in Zandipour et al. (2008), through implementation of a multi-scale grid for different regions. Grid sizes in the open sea state were made larger, making prediction coarser and in regions where vessels had limited movement the grid was made smaller allowing for a tighter degree of future prediction. This was tolerated when considered in light of the uncertainty in the underlying data distribution.

In Li et al. (2006) Support Vector Machines (SVMs) were applied to marine anomaly detection. Vessel tracks were analysed and lower level data such as position, speed and heading abstracted to higher level movement features for the track, such as turn left or loop. These features were combined with other associated information and the BIRCH clustering algorithm Zhang et al. (1996) used to form clusters of feature vectors. Cluster feature vectors describing the cluster centroid and cluster radius were then formed through measurement of the labelled feature vectors using a Euclidean distance spread statistic, and compari-

son of the value to the cluster radius. An SVM was then trained on the clustered data. New tracks were abstracted to movement motifs and classified through input to the SVM, thus following a slippage alternative hypothesis when accommodating anomalies. A SVM trained on the movement motif abstractions was demonstrated to correctly classify a significantly higher percentage of test data.
State based anomaly detectors have also previously been applied to the detection of marine anomalies. In Brax et al. (2010) discrete states were used to represent features in the data, for example a course state could consist of four states: north, south, east and west. These discrete states were combined into a composite state that captured the dependencies between atomic states. The number of occurrences and transitions into the composite state was then used to build a model of normal occurrences and transitions, following the inherent alternative hypothesis to build the anomaly free model. Subsequent tracks were identified as anomalous if the probability exceeded a user defined reference value from the anomaly free model, as per the deterministic hypothesis. This was demonstrated as being more effective than detectors based on kernel density estimators and Gaussian mixture models. State based models have also been investigated to detect anomalous port arrival. In Lane et al. (2010) port arrivals (a sequence of events) were characterised using a Markov model representing the discrete-time stochastic process, where the distribution over states (ports in this case) at a particular time step depends only on the step preceding it. Each row of the transition matrix represents the probability that a ship will move from that port to any of the other ports. Unobserved port arrivals are captured by altered probabilities in the transition matrix.
Bayesian networks have also previously been applied to detection of anomalous ship movement Mascaro et al. (2011). In this instance Bayesian networks were used to analyse both time series and track summary data thereby providing a means of detecting anomalies at both a broad and short time scale. In order to identify a sensible model of normality for the data a Bayesian network learner was then applied to the collected data, thus following an inherent alternative approach to the accommodation of anomalies. Anomalous tracks could be identified by comparison against a model of normality, thus following a deterministic hypothesis to anomaly detection. This relied on a user set threshold value, above

which a track could be identified as anomalous. It was demonstrated that the scores produced from the time series model of the track were quite distinct from those generated from the track summary model, the different models identifying different aspects of the track. It was also noted a further advantage of Bayesian networks is that they provide a causal model that a human user, such as a surveillance officer, can understand, interact with and explore.

Kalman filters for tracking are a popular approach to modelling ship dynamics and have previously been applied to track information extraction from HF RADAR data Laws et al. (2011). The Kalman filter was applied sequentially to time series data in order to extract a track of the vessels movement. At each time step the set of measurement points near the track were tested using a distance metric spread statistic and included in subsequent updates if within a reference value. One clear advantage of this approach is that ship dynamics can be encoded within the filter, thus improving future prediction. Related techniques which have previously been applied to track analysis are Gaussian processes Will et al. (2011). In this instance the Gaussian process was used to regress through a data vector of the vessels speed and location. Anomalies were detected based on a distance metric spread statistic from the mean value, points falling outside this bound were deemed anomalous. One issue with Gaussian processes is that they are typically not suitable for large data sets, this has previously been overcome by using Kd-Trees as a means of data reduction. Effectively, the latter discard data points that lie further away from the predicted mean.

2.5 Summary

Common between all the reviewed techniques in the marine domain is a lack of a principled means of incorporating outliers or extreme values within the respective modelling and anomaly detection processes. The methodologies previously applied lack a probabilistic approach to setting the threshold level. Furthermore, in most cases the threshold bound remains static which does not account for the uncertainty which may exist in regions of sparse data.

It has also been discussed how temporal anomaly detection allows anomalies that may occur at a particular temporal interval to be detected. This has been

investigated in a number of ways including applying a grid to the surveillance area and predicting the subsequent grid cell into which the vessel is to move. The problem with this approach is that many grid cells can have very few detection events, yet still be valid non-anomalous tracks, such as in open water. It also introduces the problem of selecting an appropriate division for the grid. A time series analysis of the track circumvents this issue and allows temporal anomalies to be detected in any geographic region, even one in which information is sparse. The domain review has also highlighted that little work has been undertaken in applying the power of Bayesian non-parametric methods such as Gaussian processes to the marine domain. These offer the advantage of a non-parametric approach to data modelling and also provide the capacity to incorporate vessel dynamics into the model to improve modelling accuracy. The following chapter continues the investigation into maritime anomaly detection by considering the Bayesian modelling framework and identifying techniques which could be used to improve modelling accuracy.

Chapter 3

Bayesian Modelling

Mathematical modelling provides a means whereby a simplified representation of a given system can be captured. However, if a model is to be imposed on data, then the beliefs surrounding the model structure (or the model itself) must be updated in light of new evidence. This evidence is subject to uncertainty arising from measurement noise, model uncertainty and parameter uncertainty. Accurate modelling therefore rests on the manipulation of belief given the available evidence. Probability theory provides a means of expressing the uncertainty surrounding a model and Bayesian modelling provides a methodology whereby prior belief can be incorporated into the model in a principled manner, accommodating for uncertainty at all levels.

3.1 Bayesian Probability and Methodology

Bayesian probability is built around the axioms of Cox, Jaynes (2003), which enable the Bayesian interpretation of probability to be seen as an extension of logic. This is in contrast to interpreting probability as a frequency of events as derived under Kolmogorov's axioms. Bayesian probability originates from the idea that the framework should provide a means of evaluating the logical plausibility of a proposition H (truth or falsity) in the light of proposition D, the plausibility of which being expressed as $P(H|D)$.

Under these axioms, D and H can be used as elements of Aristotelian logic,

obeying a Boolean algebra. From these axioms it is therefore possible to derive the following basic logical rules, conditioned on epistemic prior knowledge X;

$$\begin{aligned} &P(D, H|X) = P(D|H, X)\, P(H|X) = P(H|D, X)\, P(D|X), \\ &P(D, H|X) + P(\neg D, H|X) = 1, \end{aligned} \tag{3.1}$$

where $\neg D$ implies that the proposition is false. These are respectively known as the product and sum rules, from which all subsequent work can be derived. Bayesian inference for example is built around a rearrangement of the product rule,

$$P(H|D, X) = \frac{P(D|H, X)\, P(H|X)}{P(D|X)}. \tag{3.2}$$

This relationship, known as *Bayes theorem*, provides a consistent methodology to update the belief held about a hypothesis H given some new data contained within D and prior knowledge X. It states that the probability of H conditioned on D and prior knowledge X (the posterior $P(H|D, X)$) is the product of the likelihood of observing D conditioned on H and X (the likelihood $P(D|H, X)$), and the probability of H conditioned on prior knowledge X (the prior $P(H|X)$). The denominator, $P(D|X)$, is a normalising term called the marginal likelihood, or evidence.

It is important to note that the posterior remains logically consistent through division by the marginal likelihood, as it is through ensuring this logical consistency that a valid numerical value can be assigned to the plausibility of the argument. This is evinced by considering n mutual exclusive propositions $D_1 \ldots D_n$; i.e. the evidence implies that no two of them can be true simultaneously. Application of the sum rule would imply that,

$$P(D_1 + \cdots + D_n|X) = \sum_{i=1}^{n} P(D_i|X), \tag{3.3}$$

If the assumption is now made that $D_1 \ldots D_n$ are not only mutually exclusive but also exhaustive; i.e. the background information stipulates that one and only one of them must be true, then in order to meet with Cox's assumption of logical

consistency, the sum of all probabilities must equal unity,

$$\sum_{i=1}^{n} P\left(D_i|X\right) = 1. \tag{3.4}$$

In order for Equation 3.4 and Equation 3.3 to remain consistent Equation 3.3 must also sum to unity even in the case where events are not mutually exclusive; thus intuitively illustrating that logical plausibilities can be expressed on a numerical range. These degrees of plausibility can now be referred to as probability. In the instance that a distribution of these variables is expressed in terms of more than one variable i.e. $P\left(H, Y|X\right)$, the marginal distribution $P\left(H|X\right)$ can be obtained by summing for all possible values of the variable,

$$P\left(h = H|X\right) = \sum_{y} P\left(h = H, y = Y|X\right). \tag{3.5}$$

This is termed the marginalisation of variable Y, which can also be considered in the case that variables take on a continuum of values. In such cases the distribution of probabilities is expressed as a probability density function (denoted here by lowercase p),

$$p\left(h = H|X\right) = \lim_{\delta h \to 0} \frac{P\left(H \leq h < H + \delta h|X\right)}{\delta h}. \tag{3.6}$$

Given Equation 3.6 the marginalisation in Equation 3.5 can now be expressed in its integral form,

$$p\left(h = H|X\right) = \int p\left(h = H, y = Y|X\right) dy. \tag{3.7}$$

It is the assignment of numerical value to events that defines a random variable (r.v. hereafter). We note as an aside that the calculation of integrals of this form often presents many problems as calculation can prove computationally intractable.

Bayesian logic provides us a method to reconcile the probability of observed distributions with the probability of unobserved distributions and prior knowledge. Additionally its sequential application, whereby the posterior becomes the next prior, ensures prior probabilities are updated in the light of new evidence.

Bayesian inference can be applied to all variables within a system model, allowing inference to take place on a global set of unknowns. Uncertainty in the value of a variable is expressed via the probability density function, of which the controlling parameters of this distribution are known as hyperparameters. Parameters which control the hyperparameter distributions are known as hyper-hyperparameters. The Bayesian methodology is therefore concerned with the principled interpretation of uncertainty embedded within the distributions over each random variable. It thus facilitates a common-sense interpretation of statistical conclusions allowing the incorporation of prior belief into the expression of the model and for uncertainty in results to be entertained.

Bayesian learning algorithms, unlike classical estimation algorithms, do not attempt to identify "best-fit" models of the data (or similarly, make "best guess" predictions for new test inputs). Instead, they compute posterior distributions for new test inputs (or similarly, compute a posterior predictive distribution over models). These distributions provide a useful way to quantify uncertainty in model estimates, and to exploit knowledge of this uncertainty in order to make more robust predictions on new test points. Whilst frequentist approaches can express uncertainty in the model predicted value, this is expressed in terms of a confidence interval in the true value. Bayesian methods quantify this uncertainty in terms of a credibility interval. Whilst both methodologies have their merits, Bayesian uncertainty forces the implementer to be transparent about the prior assumptions that have been made to allow the uncertainty to be expressed probabilistically. Given enough data, it has been shown that the effect of the prior is generally overcome by the likelihood Diaconis and Freedman (1986). Thus Bayesian and frequentist models often converge in the limit of large data.

Choice of an appropriate prior is a contentious issue within the statistics and machine learning communities. Although it could be implied that Bayesian priors are subjective, they should not be arbitrary, therefore selection of appropriate prior distributions should be based on modelling assumptions and not from mathematical convenience.

The Bayesian methodology is particularly useful in time series analysis. In such situations there are many functions that could equally explain the observed data. This prompts the consideration of a distribution over functions. There are how-

ever infinitely many possible functions which could then be used to describe the observed data. The problem then becomes how to select the appropriate function. This selection can be driven by strong underlying prior knowledge regarding the functional form, for example in a given instance a quartic function could be used to model the underlying process. Such a model has a number of unknown parameters which are inferred as part of the modelling process, this is known as parametric modelling.

Whilst the selection of a single family of functions is a viable approach, it does not lend itself well to situations where such strong prior assumptions regarding the underlying process are not readily available. We may however be in possession of less specific domain knowledge, for example that the observations result from an underlying process which is smooth, continuous and variations in the function take place over characteristic time-scales and has typical amplitude. By adopting a non-parametric approach it is possible to work mathematically with the infinite space of all functions that have these characteristics. By considering a probability distribution over this function space, the work of modelling, explaining and forecasting the data is refined by adapting this distribution. As modelling the data in such a manner is not characterised with explicit sets of parameters that govern the functional form of the model but only the scale of the distribution (through hyperparameters which are discussed later in this chapter) they are referred to as non-parametric. The formal difference between the two methods is therefore the dimensionality of the parameter space, non-parametric models are those which consider an infinite parameter space. We next focus on a particular Bayesian non-parametric model, namely the Gaussian process (GP).

3.2 Gaussian Processes

A Gaussian process is a stochastic process Rasmussen and Williams (2006) whose realizations consist of random values associated with every point in a range of times (or of space) and in which each random variable has a Gaussian distribution. Such a process therefore exploits the marginalisation property of the Gaussian distribution. It is the marginalization property that makes working with a Gaussian process feasible: we can marginalize over the infinitely-many variables that

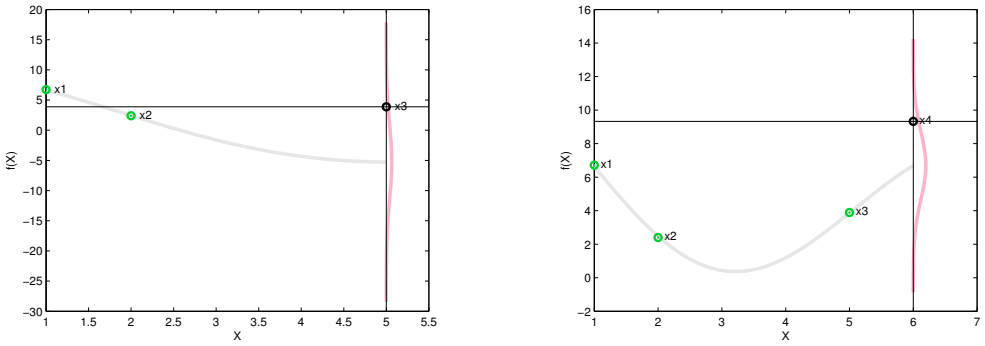
we are not interested in, or have not observed. Such processes have been used to address problems requiring classification and/or regression. Of these problems regression is concerned with modelling the relationship between a dependent variable y and one or more explanatory variables denoted x, as might be needed to capture the relationship between vessel dynamics and motion patterns. This is expressed through a standard linear regression model with Gaussian noise ϵ as,

$$y = f(x) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{3.8}$$

In practice the goal of most regression problems is to predict a new output or outputs $\boldsymbol{y}_* = y_{t+1}, \ldots y_{t+n}$ (where t is the total number of training inputs and n is the total number of predictive inputs), for some new inputs $\boldsymbol{x}_* = x_{t+1}, \ldots x_{t+n}$, conditioned on the previous output $\boldsymbol{y} = y_1, \ldots y_t$ and input $\boldsymbol{x} = x_1, \ldots x_t$ observations, along with an expression of the uncertainty in the model predicted result $p(\boldsymbol{y}_*)$, as shown in Figure 3.1. Using the product rule (Equation 3.1), this can be expressed as,

$$p(\boldsymbol{y}_*) = \int p(\boldsymbol{y}_*, \boldsymbol{y})\, d\boldsymbol{y} = \int p(\boldsymbol{y}_*|\boldsymbol{y})\, p(\boldsymbol{y})\, d\boldsymbol{y}, \tag{3.9}$$

where $\boldsymbol{y}_*$ and $\boldsymbol{y}$ are implicitly conditioned on $\boldsymbol{x}_*$ and $\boldsymbol{x}$ respectively. Many approaches fit models to the data through restriction of the class of functions i.e. quadratic functions, and learn parameters to optimise the model to the data. This approach is referred to as parametric modelling, and has the problem that it must be decided in a prior iteration from which class the model should belong. Furthermore, increasing parametric class flexibility can result in data overfitting. An alternative approach is to assign a prior probability to an entire class of functions (following the Bayesian methodology), where higher probabilities are given to functions that are considered more probable. This may appear computationally intractable; intuitively there are infinitely many possible functions. However, future realisations of a function can be considered as a vector with each entry specifying the function value $\boldsymbol{f}(\boldsymbol{x})$ for particular inputs $\boldsymbol{x}$, $\boldsymbol{f} = f(x_1) \ldots f(x_t)$. The choice of function can therefore be conditioned on the observed output $\boldsymbol{y}$, parameterised by the input $\boldsymbol{x}$ in order to infer its form, $p(\boldsymbol{f}|\boldsymbol{y})$. Inference will then give the same answer as if the infinitely many other points are ignored,



(a) (b)

Figure 3.1: Illustration of the mechanism for Gaussian process prediction. The green points are the training points and the black point is the new predictive point, against which subsequent points are conditioned $p(y_{t+1}|\boldsymbol{y})$. The predicted distribution of $p(y_{t+1}|\boldsymbol{y})$ is then shown by the pink line. In Figure 3.1a the Gaussian process is being forced to make a prediction some distance from the last observed training point. In this case the estimate is less certain of the location of the target, evinced by the large variance of the predicted Gaussian. This is in contrast to Figure 3.1b, where the prediction is a smaller distance into the future and the Gaussian process is more certain of the location of the predictive distribution.

such a property is guaranteed by the marginalisation property of the Gaussian distribution and is graphically illustrated in Figure 3.2.

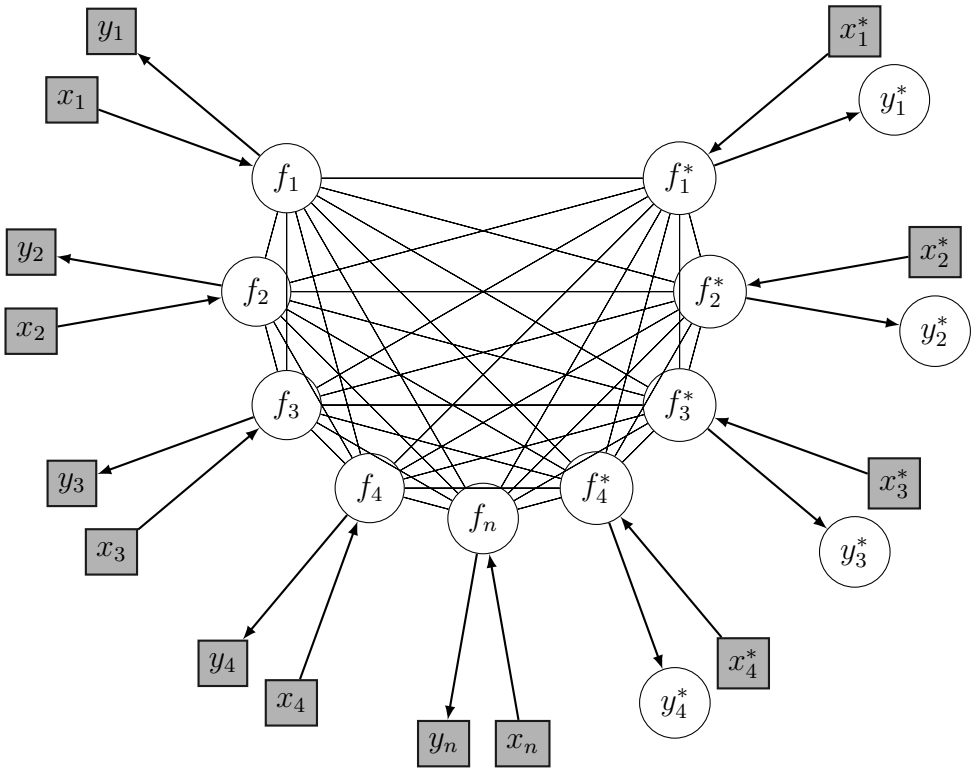


Figure 3.2: Gaussian process graphical model. Square nodes are observed (clamped), round nodes stochastic (free). All pairs of latent variables are connected. Predictions y^* depend only on the corresponding single latent f^*. Notice, that adding a triplet x_n^*, f_n^*, y_n^* does not influence the distribution. This is guaranteed by the marginalisation property of the Gaussian process. This explains why we can make inference using a finite amount of computation.

In order to evaluate Equation 3.9 it can be noted that the expression is dependent on obtaining the marginal form $p(\boldsymbol{y})$. This marginal form can be expressed, via the product rule, as the probability of observing the data given a function, combined with a prior belief as to the functions form,

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{f})\, d\boldsymbol{f}. \tag{3.10}$$

To define the form of the conditional distribution $p(\boldsymbol{y}|\boldsymbol{f})$, it can be noted from

Equation 3.8 that the noise process on the function is assumed to follow a Gaussian distribution, and is independent on each data point. Therefore the conditional distribution will be an isotropic Gaussian with a mean given by a function expressing the highest probability of fitting the data, and uncertainty expressed by the noise,

$$p\left(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{x}\right)=\mathcal{N}\left(\boldsymbol{y};\boldsymbol{f},\sigma^{2}\boldsymbol{I}_{n}\right), \tag{3.11}$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and σ^2 represents the variance of the isotropic noise. To define the form of the prior distribution $p\left(\boldsymbol{f}\right)$, the expression for linear regression, Equation 3.8, should be considered. Noting that under the assumption that noise is independent and can be added at a later stage of the derivation, this allows the model for regression to be defined as,

$$\boldsymbol{y}=\boldsymbol{f}\left(\boldsymbol{x}\right)=\boldsymbol{w}\boldsymbol{\Phi}\left(\boldsymbol{x}\right). \tag{3.12}$$

In this form the function value is expressed as a linear combination of adaptive parameters $\boldsymbol{w}$ and the design matrix $\boldsymbol{\Phi}\left(\boldsymbol{x}\right)$. The design matrix comprises the result of evaluating a basis function at each of the variables in $\boldsymbol{x}$, where the basis function provides a further means of expressing the relationship between input and output such that nonlinearities may be captured. This ensures the function is not simply restricted to a function of linear combinations of $\boldsymbol{w}$ as would be the case if $\boldsymbol{\Phi}\left(\boldsymbol{x}\right)=\boldsymbol{x}$. Under the assumption that the distribution of weights is described by a Gaussian distribution a Gaussian prior needs to be placed over the weights, $\boldsymbol{w}$.

$$p\left(\boldsymbol{w}\right)=\mathcal{N}\left(\boldsymbol{w};\boldsymbol{0},\alpha^{-1}\boldsymbol{I}\right) \tag{3.13}$$

This takes a Gaussian distribution as this allows for the weights to be positive or negative, and it is the least biasing assumption we can make, it is the maximum entropy prior. The Gaussian distribution has maximum entropy among all real-valued distributions with specified mean and variance. Therefore, the assumption of Gaussianity imposes the minimal prior structural constraint beyond these moments. Any other distribution will add bias from our choice into the

model. The distribution over the weights is assumed to have zero mean with variance governed by the hyperparameter α. By noting that $\boldsymbol{y}$ is a linear combination of Gaussian variables and $\boldsymbol{y} = \boldsymbol{w}\boldsymbol{\Phi}(\boldsymbol{x})$ it only remains to find the mean and covariance of the prior distribution over functions $p(\boldsymbol{f})$,

$$\begin{aligned} &\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\Phi}\,\mathbb{E}[\boldsymbol{w}] = \boldsymbol{0}, \\ &\operatorname{cov}[\boldsymbol{y}] = \mathbb{E}\left[\boldsymbol{y}\boldsymbol{y}^{\top}\right] = \boldsymbol{\Phi}\,\mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^{\top}\right]\boldsymbol{\Phi}^{\top} = \frac{1}{\alpha}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\top} = \boldsymbol{K}, \end{aligned} \tag{3.14}$$

where we denote two independent values as x_p and x_q and $\boldsymbol{K}$ is known as the Gram matrix which has elements,

$$K_{pq} = \kappa(x_p, x_q) = \frac{1}{\alpha}\phi(x_p)\,\phi(x_q)^{\top}, \tag{3.15}$$

in which $\kappa\left(\boldsymbol{x}, \boldsymbol{x}^{\top}\right)$ is known as the kernel function. It can also be noted that $\frac{1}{\alpha} = \mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^{\top}\right]$, as from Equation 3.13 we see that $\frac{1}{\alpha}$ controls the variance of the distribution and is hence equivalent to $\mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^{\top}\right]$. The prior distribution $p(\boldsymbol{f})$ is then given by a Gaussian whose mean is zero and whose covariance is defined by the Gram matrix $\boldsymbol{K}$, i.e.

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}; \boldsymbol{0}, \boldsymbol{K}). \tag{3.16}$$

The marginal distribution can now be found by following the results illustrated in Appendix B to integrate a Gaussian distributed conditional and prior distribution,

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{f})\,p(\boldsymbol{f})\,d\boldsymbol{f} = \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{0}, \boldsymbol{K} + \sigma^2\boldsymbol{I}\right). \tag{3.17}$$

The matrix resulting from the addition of the Gram matrix $\boldsymbol{K}$ with input noise is the Gaussian process covariance matrix $\boldsymbol{C}$,

$$\boldsymbol{C}\left(\boldsymbol{x}, \boldsymbol{x}^{\top}\right) = \boldsymbol{K} + \sigma^2\boldsymbol{I}. \tag{3.18}$$

By exploiting the results from Appendix A, it is possible to partition the covariance matrix. Therefore the joint probability of observed and predicted output

values $p(\boldsymbol{y}, \boldsymbol{y}_*)$ can be obtained through inclusion of the joint covariance,

$$\boldsymbol{C}_{1\ldots t+n} = \begin{pmatrix} \boldsymbol{C}_{1\ldots t} & \boldsymbol{k} \\ \boldsymbol{k}^\top & \boldsymbol{c} \end{pmatrix}, \tag{3.19}$$

where $\boldsymbol{c} = \kappa\left(\boldsymbol{x}_*, \boldsymbol{x}_*^\top\right) + \sigma^2 \boldsymbol{I}$ describes the correlation between a vector of new input locations. Similarly $\boldsymbol{k} = \kappa\left(\boldsymbol{x}, \boldsymbol{x}_*\right)$ describes the correlation between the new input locations and training data input locations. The results in Appendix A illustrate how a matrix can be partitioned to obtain the individual marginal distributions from the joint form. Thus performing the same operation as integrating out the $\boldsymbol{y}$ terms to give,

$$\begin{aligned} p\left(\boldsymbol{y}_*\right) &= \mathcal{N}\left(\boldsymbol{y}_*; \boldsymbol{m}\left(\boldsymbol{x}_*\right), \boldsymbol{v}\left(\boldsymbol{x}_*\right)\right), \\ \boldsymbol{m}\left(\boldsymbol{x}_*\right) &= \boldsymbol{k}^\top \boldsymbol{C}^{-1} \boldsymbol{y}, \\ \boldsymbol{v}\left(\boldsymbol{x}_*\right) &= \boldsymbol{c} - \boldsymbol{k}^\top \boldsymbol{C}^{-1} \boldsymbol{k}. \end{aligned} \tag{3.20}$$

3.2.1 The Mean Function

The mean function describes prior knowledge of the underlying function in the presence of no data; the model tending to this result in regions of uncertainty. Therefore any such prior information should be included within the mean function. However, in cases where no information is available concerning the mean of the data, a typical approach is to assume the mean is zero. This avoids any loss in generality as in our ignorance it is typical to assume that any possible trend can be increasing or decreasing; thus the model will be symmetric in the most general case.

3.2.2 The Kernel Function

The kernel function encodes the belief and assumptions about the properties of the underlying function Rasmussen and Williams (2006), as illustrated in Figure 3.3. The kernel function is composed of a combination of basis functions. Its resulting matrix, $\boldsymbol{K}$, is related to $\frac{1}{\alpha}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ in equation 3.14, so $\boldsymbol{K}$ must be positive semi-definite; restricting selection. However, there exist many possible choices for valid

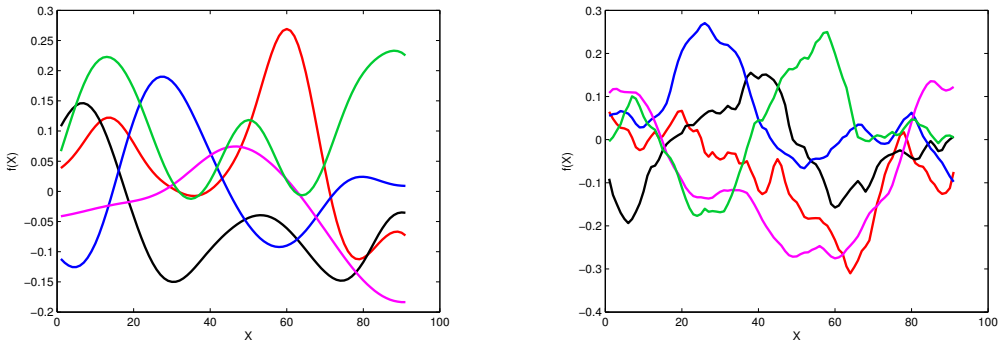
kernel functions to encode the information coupling between values. Denoting two independent values as x_p and x_q, we present three commonly used kernels as examples, the squared exponential, Matérn $\frac{3}{2}$, and the Matérn $\frac{1}{2}$ respectively,

$$\kappa_{SE}\left(x_p, x_q\right) = \sigma_0^2 \exp\left(-\frac{|x_p - x_q|^2}{2\lambda^2}\right), \tag{3.21}$$

$$\kappa_{\frac{3}{2}}\left(x_p, x_q\right) = \sigma_0^2 \left(1 + \frac{\sqrt{3}|x_p - x_q|}{\lambda}\right) \exp\left(-\frac{\sqrt{3}|x_p - x_q|}{\lambda}\right), \tag{3.22}$$

$$\kappa_{\frac{1}{2}}\left(x_p, x_q\right) = \sigma_0^2 \exp\left(-\frac{|x_p - x_q|}{\lambda}\right), \tag{3.23}$$

The hyperparameters σ_0, λ and ϵ are referred to, respectively, as the amplitude, length and noise scale. They encode the characteristics of the function, thus for our purposes the hyperparameters need to be inferred from an anomaly free training data set.



(a) (b)

Figure 3.3: Samples from a Gaussian process for a squared exponential (a) and Matérn $\frac{3}{2}$ (b) kernel. In both cases the same hyperparameter values were used. It can be observed that the samples from the squared exponential are very smooth, compared to the Matérn $\frac{3}{2}$. This is due to the squared exponential kernel being infinitely differentiable, meaning that the resultant Gaussian process has mean square derivatives of all orders. As the Matérn class is only twice differentiable, it is therefore better suited to situations in which we believe the true function exhibits rougher characteristics.

3.2.3 Multiple-Output Gaussian Process Kernel

Gaussian process regression can be extended to consider the correlation in multiple outputs through parameterisation of the covariance matrix Osborne (2010). Through unconstrained optimization, where the upper-triangular elements in the variance-covariance matrix are re-parameterised in such a way that the resulting estimate must be positive semi-definite we can discover the correlation between the selected output for the given inputs.

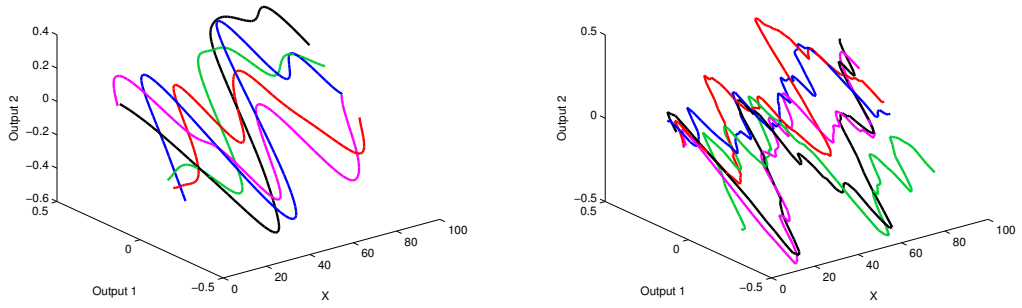
While many techniques for parameterisation of the covariance matrix exist Pinheiro and Bates (1996), the spherical parameterisation benefits from the computational efficiency of the Cholesky decomposition and allows the parameterisation to be expressed in terms of the variance and covariance between outputs. In this form it is the Cholesky factor of the output $n \times n$ covariance which is parameterised, $\boldsymbol{\Sigma} = \boldsymbol{L}^{\top}(\boldsymbol{\theta})\,\boldsymbol{L}(\boldsymbol{\theta})$. Thus it can be ensured that the final covariance matrix remains positive semi-definite. By letting $\boldsymbol{L}_i$ denote the i^{th} column of the Cholesky factorisation of $\boldsymbol{\Sigma}$ and where $\boldsymbol{l}_i$ denotes the spherical coordinates of the first i elements of $\boldsymbol{L}_i$, where $i = 2 \dots t$, we can then express the parameterisations as,

$$\begin{aligned}
[\boldsymbol{L}_1]_1 &= [\boldsymbol{l}_1]_1 \\
[\boldsymbol{L}_i]_1 &= [\boldsymbol{l}_i]_1 \cos([\boldsymbol{l}_i]_2) \\
[\boldsymbol{L}_i]_2 &= [\boldsymbol{l}_i]_1 \sin([\boldsymbol{l}_i]_2) \cos([\boldsymbol{l}_i]_3) \\
&\dots \\
[\boldsymbol{L}_i]_{i-1} &= [\boldsymbol{l}_i]_1 \sin([\boldsymbol{l}_i]_2) \dots \cos([\boldsymbol{l}_i]_i) \\
[\boldsymbol{L}_i]_i &= [\boldsymbol{l}_i]_1 \sin([\boldsymbol{l}_i]_2) \dots \sin([\boldsymbol{l}_i]_i)\,.
\end{aligned} \tag{3.24}$$

The product of the Cholesky factor parameterisation of the output spherical covariance between two outputs can therefore be expressed as,

$$\boldsymbol{\Sigma} = \begin{bmatrix} [\boldsymbol{l}_1]_1^2 & [\boldsymbol{l}_1]_1 [\boldsymbol{l}_2]_1 \cos([\boldsymbol{l}_2]_2) \\ [\boldsymbol{l}_1]_1 [\boldsymbol{l}_2]_1 \cos([\boldsymbol{l}_2]_2) & [\boldsymbol{l}_2]_2^2 \end{bmatrix}. \tag{3.25}$$

This is illustrated in Figure 3.4 for a squared exponential and Matérn $\frac{3}{2}$ kernel.



(a) (b)

Figure 3.4: Samples from a Gaussian process for a multiple output squared exponential (a) and multiple output Matérn $\frac{3}{2}$ (b) kernel. This illustrates how a Gaussian process can be used to regress across multiple outputs. It can also be observed that samples from the squared exponential Gaussian process are smoother across multiple outputs, compared to the Matérn $\frac{3}{2}$ for the same values.

In many applications we desire to extend the techniques of regression analysis in order to predict, filter or smooth through a series of data points. The next section discusses the differences between these concepts and how a Gaussian process needs to be adapted in order to perform these functions.

3.3 Prediction, Filtering and Smoothing

It is prudent at this point to note the difference between the concepts of prediction, filtering and smoothing as the Gaussian process can be used to both predict or smooth depending on the input vector passed to it. The process of prediction involves calculating the posterior distribution of output values for some time step t based on the previous observations,

$$p\left(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}\right). \tag{3.26}$$

By contrast the goal of filtering is to compute the posterior distribution of outputs for the input at x_t based on the previous observations up to a given time step t,

$$p\left(\boldsymbol{y}_t|\boldsymbol{y}_{1:t}\right). \tag{3.27}$$

Finally smoothing is the process of producing an estimate for an output at a given time step t after receiving the output measurements up to time step T, where $T > t$,

$$p\left(\boldsymbol{y}_t|\boldsymbol{y}_{1:T}\right). \tag{3.28}$$

The difference between filters and smoothers is that the filter computes its estimates using only the output measurements obtained prior to, and including time step t; the smoother uses also the measurements obtained after time step t. These differences are important to note in order to ensure accurate comparison can be made between methods discussed in the proceeding chapter.
An important set of techniques which have previously been applied to the problems of filtering, prediction, and smoothing are those based around the Kalman filter, predictor and smoother Grewal and Andrews (2008). Such methods have proven effective in predicting and smoothing navigational data Bar-Shalom et al. (2001), it would therefore be advantageous to consider whether any aspects of these methods could be incorporated into the Gaussian process to improve the modelling process. As such, the next few sections discuss the Kalman filter and smoother, before considering how aspects of Kalman based methods can be incorporated into the Gaussian process.

3.4 Kalman Filters

Many dynamic system models can be written in state space form. In such a form it is assumed that the dynamics of a system can be captured in terms of a (partially) unobserved state space, as illustrated in Figure 3.5. Kalman filter models comprise two parts; a measurement equation which relates the time series observations $\boldsymbol{y}_t$ to the system state $\boldsymbol{f}_t$ at time t. Along with a Markovian transition equation that describes the evolution of the state vector over time,

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{H}\boldsymbol{f}_t + \boldsymbol{d}_t + \boldsymbol{w}_t, \text{ where } \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}), \\ \boldsymbol{f}_{t+1} &= \boldsymbol{A}\boldsymbol{f}_t + \boldsymbol{b}_t + \boldsymbol{e}_t, \text{ where } \boldsymbol{e}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}), \end{aligned} \tag{3.29}$$

where $\boldsymbol{H}$ and $\boldsymbol{A}$ are respectively known as the output transition matrix and state transition matrix which describe the system model, determining how the current state and output vectors relate to the predicted future measurement and state vectors. The vectors $\boldsymbol{d}_t$ and $\boldsymbol{b}_t$ describe some known input i.e. system information. The variables $\boldsymbol{e}_t$ and $\boldsymbol{w}_t$ describe the uncertainty within the system model, and are controlled by system error covariances $\boldsymbol{Q}$ and $\boldsymbol{R}$.

The Kalman filter leads to an iterative process, starting from a given set of initial conditions $\hat{\boldsymbol{f}}_{1|0}$ and associated uncertainty in the estimate $\boldsymbol{P}_{1|0}$. A prediction of the system state and associated uncertainty is produced $\boldsymbol{f}_{t+1}, \boldsymbol{P}_{t+1}$. This prediction at input $t+1$ is based on the previously estimated system state information $\hat{\boldsymbol{f}}_{1\ldots t}$, and is constrained by the system model $\boldsymbol{A}$. When new measurement information is received at the predicted input, $t+1=t$, an updated estimate of the true system state and associated uncertainty can be produced, $\hat{\boldsymbol{f}}_t, \boldsymbol{P}_t$. The estimate is obtained by updating the predicted state value using the observed measurement information $\boldsymbol{y}_t$ along with a scaling factor which encompasses the system information and uncertainty. The process then repeats with the subsequent prediction using the previously estimated state in its prediction calculation. The Kalman filter therefore provides a compressed means of representing and retaining system information in the form of the system state. It does not require all previous measurement information be retained as the information is contained within the state (due to the Markovian assumption).

In deriving the Kalman filter equations it is assumed that the initial condition is Gaussian according to,

$$\boldsymbol{f}_1 \sim \mathcal{N}\left(\hat{\boldsymbol{f}}_{1|0}, \boldsymbol{P}_{1|0}\right), \tag{3.30}$$

where $\hat{\boldsymbol{f}}_{1|0}$ and $\boldsymbol{P}_{1|0}$ are the initial mean covariance conditioned on no prior information. It is typical in this instance to initialise this procedure with a mean value of 0 and a large variance. As noted in Appendix B the properties of the Gaussian ensure that the Gaussian distributional form is preserved throughout the process. Therefore it follows that, given the initial condition in Equation

3.30, the future state predictions will distributed as,

$$p\left(\boldsymbol{f}_t|\boldsymbol{y}_{1:t}\right) = \mathcal{N}\left(\boldsymbol{f}_t; \hat{\boldsymbol{f}}_{t|t-1}, \boldsymbol{P}_{t|t-1}\right). \tag{3.31}$$

From Equation 3.29 and the results in Appendix B it then follows that,

$$\begin{aligned} p\left(\boldsymbol{y}_t|\boldsymbol{f}_t, \boldsymbol{y}_{1:t-1}\right) = p\left(\boldsymbol{y}_t|\boldsymbol{f}_t\right) = \mathcal{N}\left(\boldsymbol{y}_t; \boldsymbol{H}\boldsymbol{f}_t + \boldsymbol{d}_t, \boldsymbol{R}\right), \\ p\left(\boldsymbol{f}_t|\boldsymbol{y}_{1\ldots t}\right) = \mathcal{N}\left(\boldsymbol{f}_t; \hat{\boldsymbol{f}}_{t|t}, \boldsymbol{P}_{t|t}\right), \text{ where } \boldsymbol{P}_{t|t} = \boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{H}\boldsymbol{P}_{t|t-1}. \end{aligned} \tag{3.32}$$

where $\boldsymbol{K}_t$ is the scaling factor (termed the Kalman gain) given by,

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{H}^\top\boldsymbol{S}_t^{-1} \text{ and } \boldsymbol{S}_t = \boldsymbol{H}\boldsymbol{P}_{t|t-1}\boldsymbol{H}^\top + \boldsymbol{R}. \tag{3.33}$$

The function estimate at time t is given by,

$$\begin{aligned} \hat{\boldsymbol{f}}_{t|t} &= \boldsymbol{P}_{t|t}\left(\boldsymbol{H}^\top\boldsymbol{R}^{-1}\left(\boldsymbol{y}_t - \boldsymbol{d}_t\right) + \left(\boldsymbol{P}_{t|t-1}\right)^{-1}\hat{\boldsymbol{f}}_{t|t-1}\right), \\ &= \left(\boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{H}\boldsymbol{P}_{t|t-1}\right)\boldsymbol{H}^\top\boldsymbol{R}^{-1}\left(\boldsymbol{y}_t - \boldsymbol{d}_t\right) + \boldsymbol{P}_{t|t}\left(\boldsymbol{P}_{t|t-1}\right)^{-1}\hat{\boldsymbol{f}}_{t|t-1}, \\ &= \underbrace{\left(\boldsymbol{P}_{t|t-1}\boldsymbol{H}^\top\right)}_{=\boldsymbol{K}_t\boldsymbol{S}_t} - \boldsymbol{K}_t\underbrace{\boldsymbol{H}\boldsymbol{P}_{t|t-1}\boldsymbol{H}^\top}_{=\boldsymbol{S}_t-\boldsymbol{R}}\boldsymbol{R}^{-1}\left(\boldsymbol{y}_t - \boldsymbol{d}_t\right) + \boldsymbol{P}_{t|t}\left(\boldsymbol{P}_{t|t-1}\right)^{-1}\hat{\boldsymbol{f}}_{t|t-1}, \\ &= \boldsymbol{K}_t\left(\boldsymbol{S}_t - \boldsymbol{S}_t + \boldsymbol{R}\right)\boldsymbol{R}^{-1}\left(\boldsymbol{y}_t - \boldsymbol{d}_t\right) + \left(\boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{H}\boldsymbol{P}_{t|t-1}\right)\left(\boldsymbol{P}_{t|t-1}\right)^{-1}\hat{\boldsymbol{f}}_{t|t-1}, \\ &= \boldsymbol{K}_t\left(\boldsymbol{y}_t - \boldsymbol{d}_t\right) + \left(\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}\right)\hat{\boldsymbol{f}}_{t|t-1}, \\ &= \hat{\boldsymbol{f}}_{t|t-1} + \boldsymbol{K}_t\left(\boldsymbol{y}_t - \boldsymbol{d}_t - \boldsymbol{H}\hat{\boldsymbol{f}}_{t|t-1}\right). \end{aligned} \tag{3.34}$$

The function forecast distribution is then given by,

$$\begin{aligned} p\left(\boldsymbol{f}_{t+1}|\boldsymbol{f}_t, \boldsymbol{y}_{1:t}\right) = p\left(\boldsymbol{f}_{t+1}|\boldsymbol{f}_t\right) = \mathcal{N}\left(\boldsymbol{f}_{t+1}; \boldsymbol{A}\boldsymbol{f}_t + \boldsymbol{b}_t, \boldsymbol{Q}\right), \\ p\left(\boldsymbol{f}_{t+1}|\boldsymbol{y}_{1:t}\right) = \mathcal{N}\left(\boldsymbol{f}_{t+1}; \hat{\boldsymbol{f}}_{t+1|t}, \boldsymbol{P}_{t+1|t}\right), \end{aligned} \tag{3.35}$$

where,

$$\begin{aligned} \hat{\boldsymbol{f}}_{t+1|t} &= \boldsymbol{A}\hat{\boldsymbol{f}}_{t|t} + \boldsymbol{b}_t, \\ \boldsymbol{P}_{t+1|t} &= \boldsymbol{A}\boldsymbol{P}_{t|t}\boldsymbol{A}^\top + \boldsymbol{Q}. \end{aligned} \tag{3.36}$$

We note that the filtering process is reliant on the system model in its calculation, therefore the modelling process is constrained by this form. As such the Kalman filter is a parametric technique.

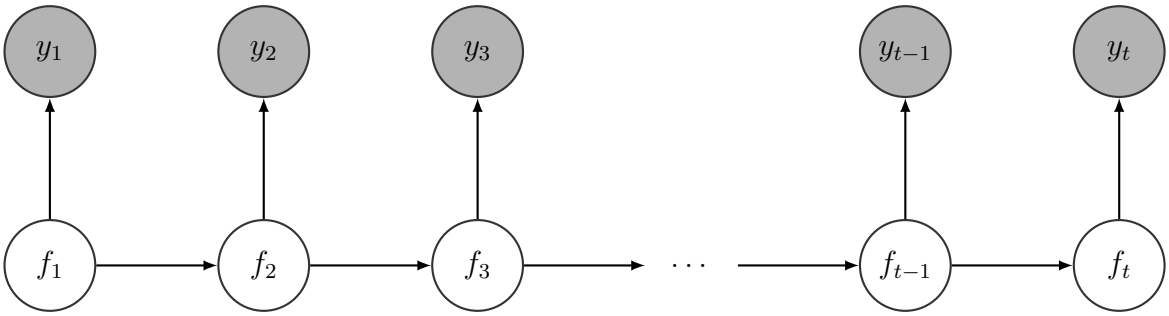


Figure 3.5: In the Kalman filter the $\boldsymbol{f}_t$ state variables form a Markov chain that is unobserved or hidden, observations $\boldsymbol{y}_t$ at each time t are used to update the current estimate of the model state $\hat{\boldsymbol{f}}_{t|t}$. Shaded nodes indicate variables with observed values.

3.5 Kalman Smoothers

The discrete-time Kalman smoother, also known as the Rauch-Tung-Striebel (RTS) smoother, can be used for computing the smoothing solution for the Kalman filter. In the smoothing step all the measurements are reprocessed after the last measurement has been made and the filtering step has been completed. The smoothed mean $\hat{\boldsymbol{f}}_t^s$ and covariance $\boldsymbol{P}_t^s$ are calculated with the following equa-

tions,

$$\begin{aligned}
\hat{\boldsymbol{f}}_{t+1|t} &= \boldsymbol{A}\hat{\boldsymbol{f}}_{t|t} \\
\boldsymbol{P}_{t+1|t} &= \boldsymbol{A}\boldsymbol{P}_{t|t}\boldsymbol{A}^{\top} + \boldsymbol{Q} \\
\boldsymbol{C} &= \boldsymbol{P}_{t|t}\boldsymbol{A}^{\top}\left(\boldsymbol{P}_{t+1|t}\right)^{-1} \\
\hat{\boldsymbol{f}}^{s}_{t|t} &= \hat{\boldsymbol{f}}_{t|t} + \boldsymbol{C}\left(\boldsymbol{f}^{s}_{t+1|t} - \hat{\boldsymbol{f}}_{t+1|t}\right) \\
\boldsymbol{P}^{s}_{t|t} &= \boldsymbol{P}_{t|t} + \boldsymbol{C}\left(\boldsymbol{P}^{s}_{t+1|t} - \boldsymbol{P}_{t+1|t}\right)\boldsymbol{C}^{\top}
\end{aligned} \tag{3.37}$$

The Kalman smoother performs the same smoothing operation as a Gaussian process which is passed all the output measurements up to a time step T. The two smoothing methods are used in Chapter 4 where we investigate the transfer of techniques between the two methods. We begin this investigation by first considering the reconciliation of Gaussian processes with Kalman methods in the following section.

3.6 Gaussian Processes and Kalman Methods

From the previous sections on Gaussian processes and Kalman methods it is possible to note several key differences between the methodologies. Kalman methods incorporate a noise model for both the input observation and state transition, given by the covariances $\boldsymbol{R}$ and $\boldsymbol{Q}$ respectively. Gaussian processes conversely only incorporate a noise model for the input observation, $\sigma^2\boldsymbol{I}_n$. Furthermore Kalman methods, due to their Markovian representation of the system state, efficiently provide a compressed representation of the system. By contrast the Gaussian process covariance matrix rapidly becomes computationally demanding to invert. However the Gaussian process does offer the advantage that it does not require a discretised representation of the environment in the form of a systems model, and due to its non-parametric form offers an increased flexibility to modelling problems.

To explore the relationship further, the Kalman equations for the error covariance and estimate (Equations 3.32 and 3.34) are now conditioned on a series of obser-

vations $\boldsymbol{y} = y_1 \dots y_t$ at inputs $\boldsymbol{x} = x_1 \dots x_t$, and used to predict the output values at input locations $\boldsymbol{x}_* = x_{t+1} \dots x_{t+n}$. For clarity the notation $\boldsymbol{x}' = x_1 \dots x_{t+n}$ is introduced. Making the assumption that input noise is uncorrelated, $\boldsymbol{R} = \sigma^2 \boldsymbol{I}$, the Kalman equations take the following form,

$$\begin{aligned} \hat{\boldsymbol{f}}_{t+1|t}\left(\boldsymbol{x}'\right) &= \hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}'\right) + \boldsymbol{K}_t\left(\boldsymbol{y}_t - \boldsymbol{d}_t - \boldsymbol{H}\hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}'\right)\right), \\ \boldsymbol{P}_{t+1|t}\left(\boldsymbol{x}', \boldsymbol{x}'\right) &= \boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}', \boldsymbol{x}'\right) - \boldsymbol{K}_t\boldsymbol{H}\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}', \boldsymbol{x}'\right), \\ \boldsymbol{K}_t &= \boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}', \boldsymbol{x}'\right)\boldsymbol{H}^\top\left(\boldsymbol{H}\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}', \boldsymbol{x}'\right)\boldsymbol{H}^\top + \sigma^2\boldsymbol{I}\right)^{-1}. \end{aligned} \tag{3.38}$$

Introducing the indicator matrix $\boldsymbol{G}_*$ such that $\boldsymbol{x}_* = \boldsymbol{G}_*\boldsymbol{x}'$. Pre-multiplication by the indicator matrix $\boldsymbol{G}_*$,

$$\begin{aligned} \hat{\boldsymbol{f}}_{t+1|t}\left(\boldsymbol{x}_*\right) &= \boldsymbol{G}_*\left(\hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}'\right) + \boldsymbol{K}_t\left(\boldsymbol{y}_t - \boldsymbol{d}_t - \boldsymbol{H}\hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}'\right)\right)\right), \\ \boldsymbol{P}_{t+1|t}\left(\boldsymbol{x}_*, \boldsymbol{x}_*\right) &= \boldsymbol{G}_*\left(\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}', \boldsymbol{x}'\right) - \boldsymbol{K}_t\boldsymbol{H}\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}', \boldsymbol{x}'\right)\right), \end{aligned} \tag{3.39}$$

and noting that,

$$\hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}\right) = \boldsymbol{H}\hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}'\right) \text{ and } \boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}, \boldsymbol{x}\right) = \boldsymbol{H}\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}', \boldsymbol{x}'\right)\boldsymbol{H}^\top, \tag{3.40}$$

allows us to recover the Kalman equations for $\boldsymbol{x}_*$ leading to,

$$\begin{aligned} \hat{\boldsymbol{f}}_{t+1|t}\left(\boldsymbol{x}_*\right) =& \hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}_*\right) + \boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}_*, \boldsymbol{x}\right)\left(\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}, \boldsymbol{x}\right) + \sigma^2\boldsymbol{I}\right)^{-1} \times \\ & \left(\boldsymbol{y}_t - \boldsymbol{d}_t - \hat{\boldsymbol{f}}_{t|t-1}\left(\boldsymbol{x}\right)\right), \\ \boldsymbol{P}_{t+1|t}\left(\boldsymbol{x}_*, \boldsymbol{x}_*\right) =& \boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}_*, \boldsymbol{x}_*\right) - \boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}_*, \boldsymbol{x}\right)\left(\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}, \boldsymbol{x} + \sigma^2\boldsymbol{I}\right)\right)\boldsymbol{P}_{t|t-1}\left(\boldsymbol{x}, \boldsymbol{x}_*\right). \end{aligned} \tag{3.41}$$

Under these assumptions the Gaussian process kernel is equivalent to the Kalman prior covariance Reece and Roberts (2010), and it is therefore possible for Kalman system models to be embedded within the Gaussian process, or conversely for Gaussian process kernels to be embedded within Kalman methods. Incorpora-

tion of a Kalman systems model into the Gaussian process offers the benefit that the Gaussian process can take advantage of the wide range of parametric models typically used with the Kalman filter framework. Thus it is possible to enhance modelling accuracy within the Gaussian process through incorporation of an appropriate systems model. However, this is at the expense of the introduction of more model parameters, increasing the computational demands for inference.

3.7 Summary

This chapter has discussed Bayesian modelling and provided most of the underlying theory for the following chapters. In particular the mathematical theory pertaining to Gaussian processes and Kalman filters has been discussed. It was also shown how under certain conditions the Gaussian process and Kalman filter are equivalent, thus allowing for the transfer of techniques between the two methods. A comparison and further detail on this is presented in Chapter 4, where we consider the incorporation of the Kalman near constant velocity and acceleration system models into the Gaussian process model.

Chapter 4

Modelling Vessel Dynamics

In the previous chapter we considered how the kernel function of a GP can describe the intrinsic information coupling between data points separated in space and time. This thesis is concerned with vessel track modelling, therefore the kernel should conform to known vessel dynamics wherever possible. In the Kalman filter literature two main models are employed to capture such motion patterns: the near-constant velocity and acceleration models. As the Kalman filter can be considered a special case of the Gaussian process, it is therefore possible to formulate Kalman filter models as Gaussian process kernels and hence embed a bespoke systems model within the Gaussian process. The derivation of the appropriate kernel functions is given in the next subsections.

4.1 The Near Constant Velocity Model

The near constant velocity model provides a general expression for all first-order differential functions f which are subject to a random diffusion e_t. It may be expressed as,

$$f(x_t) = f(x_{t-1}) + (x_t - x_{t-1}) f'(x_{t-1}) + e_t, \tag{4.1}$$

where the time variable x_t increases in value with increasing index k (i.e. $x_t > x_k$

if and only if $t > k$). The state vector can therefore be defined as,

$$\boldsymbol{\phi}_t = \begin{pmatrix} f(x_t) \\ f'(x_t) \end{pmatrix}, \tag{4.2}$$

and the temporal difference $\delta_t = x_t - x_{t-1} > 0$. The state update equations are hence,

$$\begin{aligned} \boldsymbol{\phi}_t &= \boldsymbol{A}_t \boldsymbol{\phi}_{t-1} + e_t, \\ \bar{\boldsymbol{\phi}}_{\boldsymbol{t}} &= \boldsymbol{A}_t \bar{\boldsymbol{\phi}}_{t-1} \text{ where } \boldsymbol{A}_t = \boldsymbol{A}(\delta_t) = \begin{pmatrix} 1 & \delta_t \\ 0 & 1 \end{pmatrix}. \end{aligned} \tag{4.3}$$

The random diffusion e_t, is defined by its covariance $\boldsymbol{Q}_t$, which can be expressed as,

$$\begin{aligned} \boldsymbol{Q}_t = \boldsymbol{Q}(\delta_t) &= \mathbb{E}[e_t e_t^\top] \\ &= \mathbb{E}\left[\left(\int_0^{\delta_t} \begin{bmatrix} \delta_t - u \\ 1 \end{bmatrix} e(t_k + u) du \right) \left(\int_0^{\delta_t} \begin{bmatrix} \delta_t - v \\ 1 \end{bmatrix} e(x_t + v) dv \right)^\top \right], \\ &= \int_0^{\delta_t} \int_0^{\delta_t} \begin{bmatrix} \delta_t - u \\ 1 \end{bmatrix} \begin{bmatrix} \delta_t - v & 1 \end{bmatrix} \mathbb{E}[e(x_t + u) e(x_t + v)^\top] du dv, \\ &= \int_0^{\delta_t} \begin{bmatrix} (\delta_t - u)^2 & (\delta_t - u) \\ (\delta_t - u) & 1 \end{bmatrix} q(x_t + u) du, \\ &\approx q \begin{bmatrix} \frac{\delta_t^3}{3} & \frac{\delta_t^2}{2} \\ \frac{\delta_t^2}{2} & \delta_t \end{bmatrix}. \end{aligned} \tag{4.4}$$

and that since q is constant it can for convenience be taken outside of $\boldsymbol{Q}_t$. The latter is therefore defined as,

$$\boldsymbol{Q}_t = \begin{bmatrix} \frac{\delta_t^3}{3} & \frac{\delta_t^2}{2} \\ \frac{\delta_t^2}{2} & \delta_t \end{bmatrix}. \tag{4.5}$$

It is now possible to derive a covariance function, κ, equivalent to the near con-

stant velocity model of the Kalman filter. From Equation 4.3 the state update equations can be expressed as a covariance,

$$\begin{aligned}\kappa(\phi_t, \phi_t) &= \boldsymbol{A}_t \kappa(\phi_{t-1}, \phi_t) \boldsymbol{A}_t^\top + q\boldsymbol{Q}_t, \\ \kappa(\phi_t, \phi_k) &= \boldsymbol{A}_t \kappa(\phi_{t-1}, \phi_k).\end{aligned} \tag{4.6}$$

These expressions for the covariance, at time step t, and the covariance between time steps t and k can alternatively be expressed as,

$$\begin{aligned}\kappa(\phi_t, \phi_t) &= \boldsymbol{A}(x_t - x_0)\kappa(\phi_0, \phi_0)\boldsymbol{A}(x_t - x_0)^\top + q\boldsymbol{Q}(x_t - x_0), \\ \kappa(\phi_t, \phi_k) &= \boldsymbol{A}(x_t - x_k)\kappa(\phi_k, \phi_k), \\ &= \boldsymbol{A}(x_t - x_k)[\boldsymbol{A}(x_k - x_0)\kappa(\phi_0, \phi_0)\boldsymbol{A}(x_k - x_0)^\top + q\boldsymbol{Q}(x_k - x_0)], \\ &= \boldsymbol{A}(x_t - x_0)\boldsymbol{A}(\phi_0, \phi_0)\boldsymbol{A}(x_k - x_0)^\top + q\boldsymbol{A}(x_t - x_k)\boldsymbol{Q}(x_k - x_0).\end{aligned} \tag{4.7}$$

The following vectors are now introduced,

$$\begin{aligned}\boldsymbol{M}(\delta) &= \begin{pmatrix} 1 & \delta \end{pmatrix}, \\ \boldsymbol{N}(\delta) &= \begin{pmatrix} \frac{\delta_t^3}{3} & \frac{\delta^2}{2} \end{pmatrix}.\end{aligned} \tag{4.8}$$

The covariance functions in Equation 4.7 can now be combined and the near constant velocity model Gaussian process kernel κ_{NCVM} expressed as,

$$\kappa_{NCVM} = \boldsymbol{M}(x_t - x_0)\kappa(\phi_0, \phi_0)\boldsymbol{M}(x_k - x_0)^\top + q\boldsymbol{M}(x_t - x_k)\boldsymbol{N}(x_k - x_0)^\top. \tag{4.9}$$

Samples drawn from the near constant velocity model kernel (Equation 4.9) are shown in Figure 4.1.

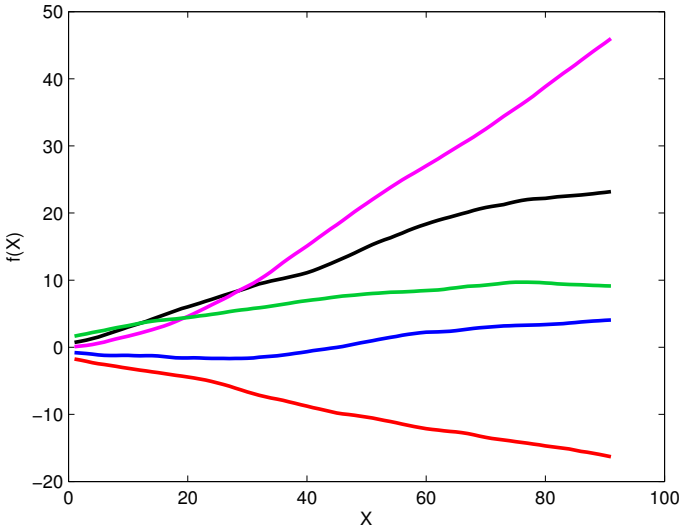


Figure 4.1: Samples drawn from the near constant velocity model kernel. The smooth draws indicate the form that the output function can take assuming a near constant velocity. The output function is constrained by the imposed physical dynamics and the resulting regression is much more constrained than the squared exponential.

4.2 The Near Constant Acceleration Model

The near constant acceleration model provides a general expression for all second-order differential functions f which are subject to a random diffusion e_t and is expressed as,

$$f(x_t) = f(x_{t-1}) + (x_t - x_{t-1}) f'(x_{t-1}) + \frac{1}{2}(x_t - x_{t-1})^2 f''(x_{t-1}) + e_t, \quad (4.10)$$

where as before the time variable x_t increases in value with increasing index t. The state vector is now defined as,

$$\boldsymbol{\phi}_t = \begin{pmatrix} f(x_t) \\ f'(x_t) \\ f''(x_t) \end{pmatrix}, \quad (4.11)$$

and the state transition matrix as,

$$\boldsymbol{A}_t = \boldsymbol{A}(\delta_t) = \begin{pmatrix} 1 & \delta_t & \frac{\delta_t^2}{2} \\ 0 & 1 & \delta_t \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.12}$$

The process noise covariance now takes the form,

$$\boldsymbol{Q}_t = \boldsymbol{Q}(\delta_t) = \begin{pmatrix} \frac{\delta_t^5}{20} & \frac{\delta_t^4}{8} & \frac{\delta_t^3}{6} \\ \frac{\delta_t^4}{8} & \frac{\delta_t^3}{3} & \frac{\delta_t^2}{2} \\ \frac{\delta_t^3}{6} & \frac{\delta_t^2}{2} & \delta_t \end{pmatrix}. \tag{4.13}$$

Following the same process as used in the derivation for the near constant velocity model the vectors $\boldsymbol{M}$ and $\boldsymbol{N}$ are defined as,

$$\begin{aligned} \boldsymbol{M}(\delta) &= \begin{pmatrix} 1 & \delta & \frac{\delta^2}{2} \end{pmatrix}, \\ \boldsymbol{N}(\delta) &= \begin{pmatrix} \frac{\delta_t^5}{20} & \frac{\delta_t^4}{8} & \frac{\delta_t^3}{6} \end{pmatrix}. \end{aligned} \tag{4.14}$$

Again, noting that the time indices should be ordered, the near constant acceleration Gaussian process kernel κ_{NCAM} can be defined as,

$$\kappa_{NCAM} = \boldsymbol{M}(x_t - x_0)\kappa(\phi_0, \phi_0)\boldsymbol{M}(x_k - x_0)^\top + q\boldsymbol{M}(x_t - x_k)\boldsymbol{N}(x_k - x_0)^\top. \tag{4.15}$$

Samples drawn from the near constant acceleration model kernel (Equation 4.15) are shown in Figure 4.2.

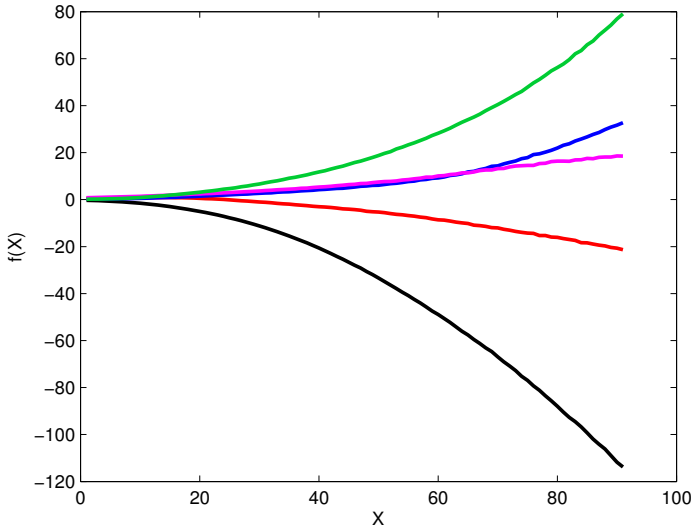


Figure 4.2: Samples drawn from the near constant acceleration model. The smooth draws indicate the form that the output function can take assuming a near constant acceleration. The output function is constrained by the imposed physical dynamics and the resulting regression is much more constrained than the squared exponential.

We next empirically validate the derived kernels through comparison of the Gaussian process kernels to the original Kalman smoother models. This leads into an evaluation of which kernel is most suited to the modelling of vessel dynamics.

4.3 Validation and Comparison of Derived Kernels

To evaluate the suitability of the derived kernels for maritime data modelling, the kernels were applied to real world vessel track data. A vessel track consisting of time stamped latitudinal and longitudinal co-ordinates of a vessels position for a given interval of time. This data was collected through a custom written web scraping application, used to collect freely available information (the code for which is given in Appendix C) and consisted of approximately 100 vessel tracks. To reduce the required processing time per track (inference taking approximately 3 to 4 seconds per track in the bivariate target space compared to only 0.5 to

1 second per track in the univariate target space on an intel core i5 2.4GHz CPU with 8GB RAM) the dimensionality of the data was reduced, allowing time stamped latitudinal and longitudinal co-ordinate data to be modelled in the univariate domain. This feature space representation of the data is discussed next.

4.3.1 Vessel Track Feature Space

The data was converted to a univariate feature space in which the first received data point is considered as the beginning of the vessel track. All subsequent data points are related to it by computing both the distance and time taken from this originating sample point. In order to take into account the approximated spherical geometry of the earth's surface we calculate this distance by application of the Haversine formula,

$$\begin{aligned} A &= \cos\phi_s \cos\phi_f \\ \Delta\hat{\sigma} &= \arctan\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + A\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right). \end{aligned} \tag{4.16}$$

where ϕ_s and ϕ_f are the latitude of two points and $\Delta\lambda$ and $\Delta\phi$ are their differences in longitude and latitude respectively. This choice of feature space has the advantage of converting the GPS information into a 1D feature vector, reducing the computational demands of processing the data. Also, the arc length between points d for a sphere of radius r and $\Delta\hat{\sigma}$ is given in radians by,

$$d = r\Delta\hat{\sigma}. \tag{4.17}$$

In the following sections the methods are applied to data converted by this described feature space representation, which essentially maps the bivariate data into a univariate distance from an origin. The track is also normalised using standard score normalisation for the individual track data, this has the effect of removing the unit of distance from the track. All graph axis using this feature space are therefore referred to as normalised distance. In the converted feature

space the unit of distance does not matter as we are only interested in modelling and detecting deviations relative to the individual track.

4.3.2 Vessel Track Modelling

A single track from the available data was randomly selected, and its dimensionality reduced. To obtain a mean value and its associated uncertainty the Gaussian process and Kalman smoother models were then passed the dimensionally-reduced track data. A point estimate of the track was used to learn the hyperparameters of the Gaussian process, it should be noted that a "strongly Bayesian" alternative to this approach does exist, namely placing distributions over the Gaussian process hyperparameters. In each model the hyperparameter values used in the Gaussian process kernel were used as the parameter values in the Kalman smoother equivalent model, for which the inferred noise values were 5.3780×10^{-9} and 26.1×10^{-3} for the constant velocity and acceleration models respectively.

The results of applying the Gaussian process and Kalman smoother constant velocity model to the data are illustrated in Figure 4.3, and the Gaussian process and Kalman smoother implementation of the constant acceleration model illustrated in Figure 4.5. The outputs from the models can be directly compared in Figures 4.4 and 4.6.

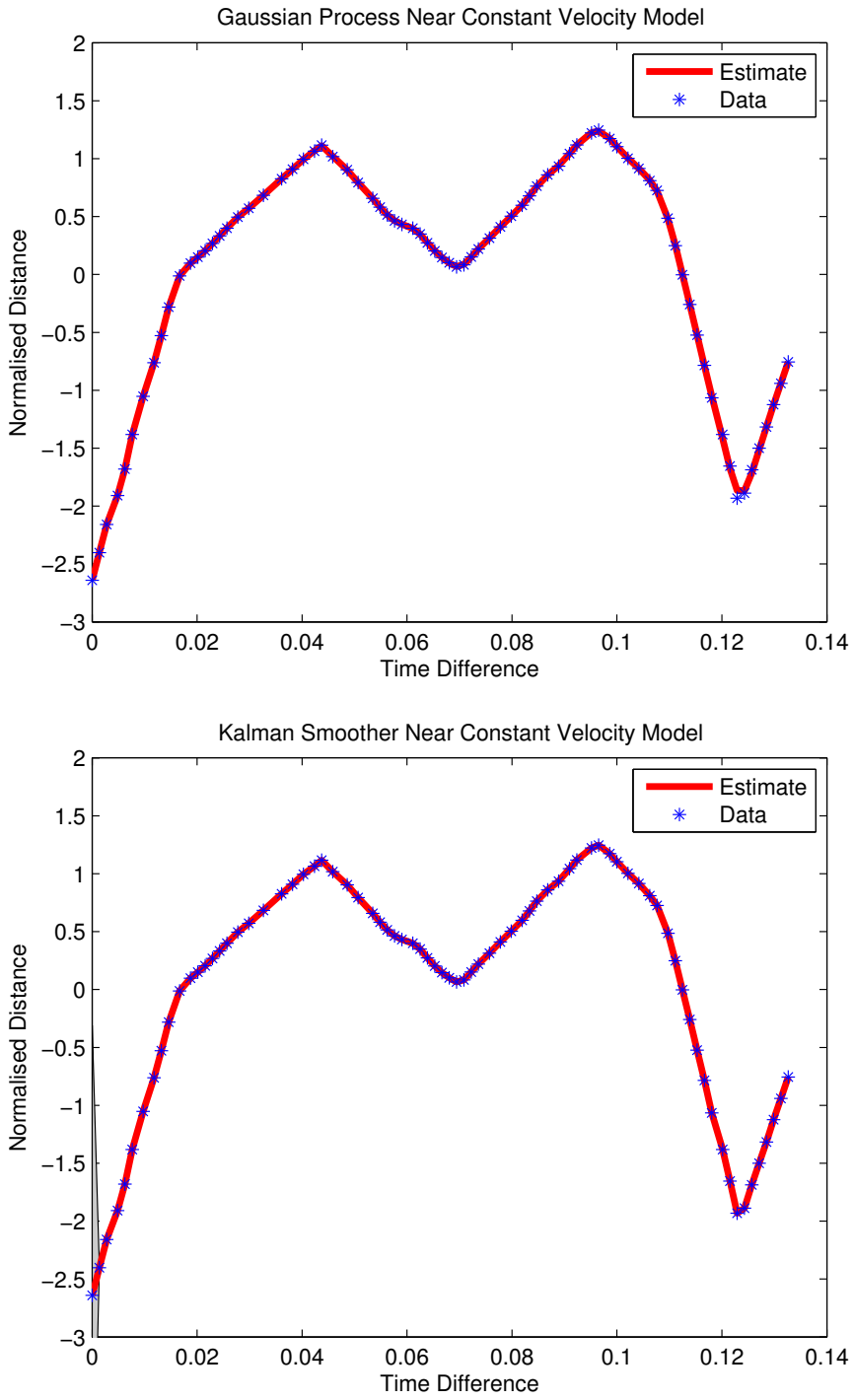


Figure 4.3: The Gaussian process (a) and Kalman smoother (b) near constant velocity model applied to vessel track co-ordinates transformed by the feature space.

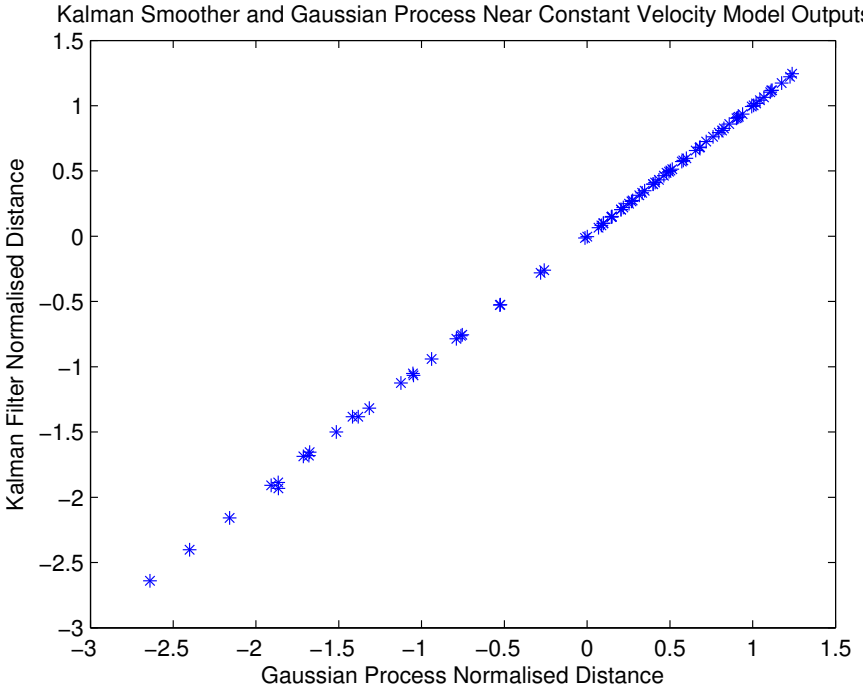


Figure 4.4: Gaussian process near constant velocity model output plotted against the output from the equivalent Kalman smoother near constant velocity model.

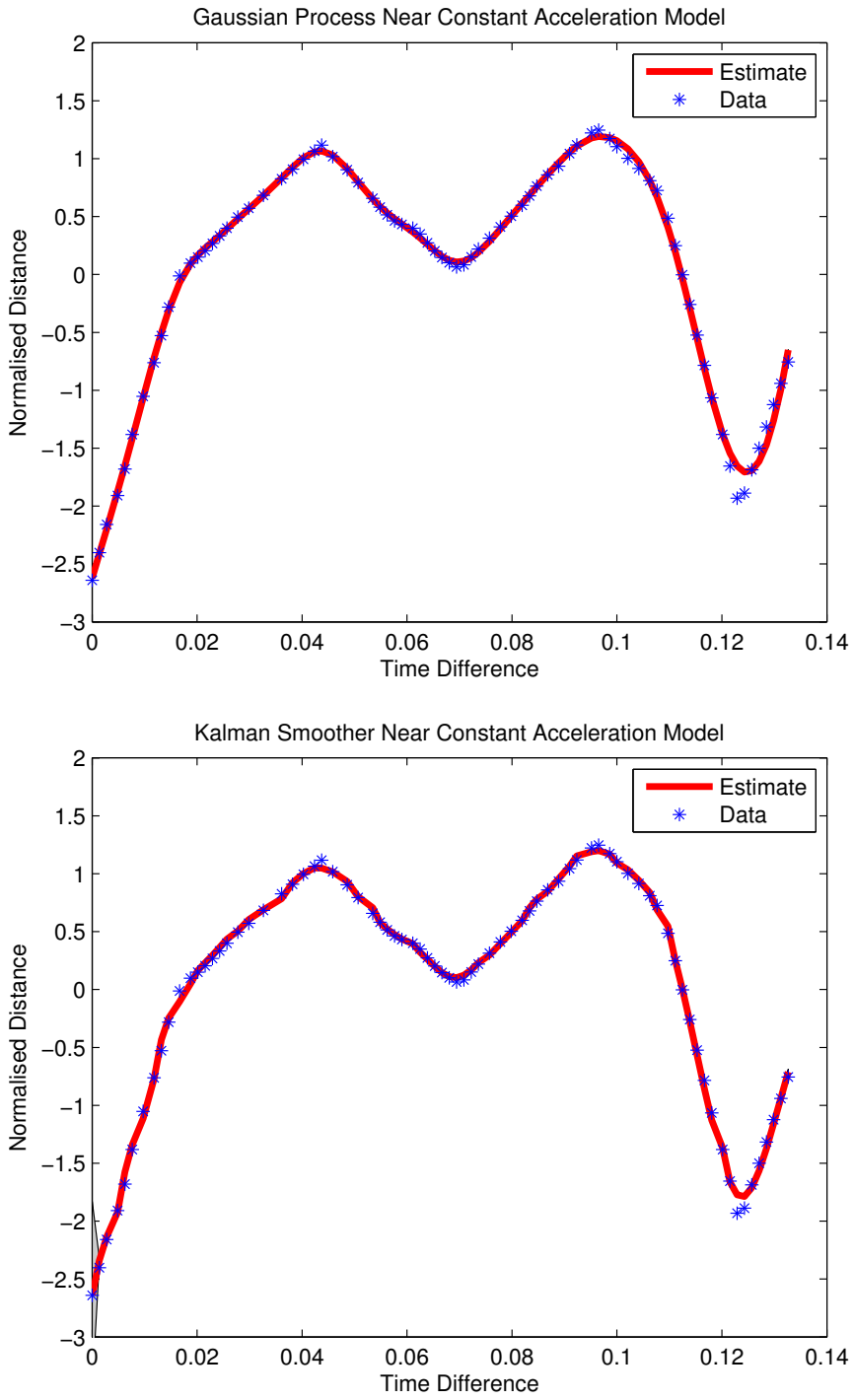


Figure 4.5: The Gaussian process (a) and Kalman smoother (b) near constant acceleration model applied to vessel track co-ordinates transformed by the feature space.

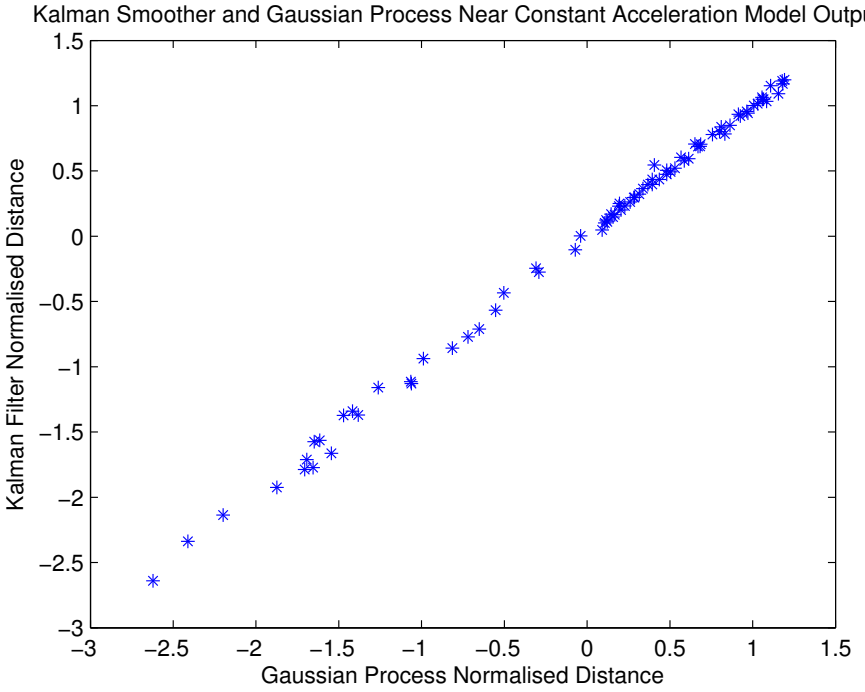


Figure 4.6: Gaussian process near constant acceleration model output plotted against the output from the equivalent Kalman smoother near constant acceleration model.

Comparing the results from the Kalman smoother and Gaussian process implementations of the near constant velocity model (Figure 4.3), the results show a high degree of similarity as illustrated in Figure 4.4. Comparatively several noticeable differences between the Kalman smoother and Gaussian process implementations of the near constant acceleration model (Figure 4.3) are illustrated in Figure 4.6. To further demonstrate the similarity between the Kalman smoother and Gaussian process near constant velocity and acceleration models the methods were applied to 30 tracks. The coefficient of variation for the correlation between the near constant acceleration model implementations was found to be 0.03 and 0.73 for the constant velocity model.

Whilst theoretically the results of the two models should align, practically there are differences in how the methods are implemented. Within the Gaussian process and Kalman smoother it is common practise to add jitter in order to prevent conditioning errors such as scaling errors due to computational implementation.

This is the practice of adding noise (a typical value being 10^{-6}) to the diagonal elements of a matrix and can be defined (for some arbitrary matrix J) as,

$$J + 10^{-6}\delta_{xy} \tag{4.18}$$

where δ_{xy} is known as the Kronecker delta and is defined as,

$$\delta_{xy} = \begin{cases} 0 \text{ for } i \neq j \\ 1 \text{ for } i = j \end{cases} \tag{4.19}$$

The practise of adding jitter therefore dilutes the informativeness of our data. This occurs to the diagonal of the Gaussian process covariance matrix and to the Kalman filter state estimate. It also occurs within different stages of the model implementation, in the Kalman smoother for example the jitter is added to the predictive covariance $\boldsymbol{P}$ as we iterate back through an estimate of the data, contrastingly this is added to the Gaussian process Gram matrix $\boldsymbol{K}$ before any estimation takes place. This practice introduces further errors to the model, and consequently a difference between the Gaussian process and Kalman models.
In order to evaluate the derived kernels, they need to be compared against other commonly used Gaussian process kernels. This is undertaken in the next section, where kernels are evaluated in order to discover the optimum kernel for maritime data modelling.

4.4 Choice of Kernel Function for Vessel Modelling

The choice of kernel function is central to accurate data modelling, and hence providing the most accurate model of vessel dynamics. To determine the optimal kernel function the performance of the squared exponential kernel (Equation 3.21), Matérn$\frac{3}{2}$ kernel (Equation 3.22), Matérn$\frac{1}{2}$ kernel (Equation 3.23) and near constant velocity kernel (Equation 4.9) were investigated. Due to a desire to implement the algorithm in an online setting the near constant acceleration kernel (Equation 4.15) was omitted from the investigation, due to the nine kernel hyperparameters, which were found to be computationally demanding to infer. This is

in contrast to the near constant velocity model which requires six hyperparameters or the squared exponential, Matérn$\frac{3}{2}$ and Matérn$\frac{1}{2}$ which only require three hyperparameter values to be inferred.

Training data consisting of 30 vessel tracks was used to estimate the GP kernel function parameters, through maximisation of the log marginal likelihood of the data Rasmussen and Williams (2006),

$$\log\left(p(\boldsymbol{y}|\boldsymbol{x})\right) = -\frac{1}{2}\log\left(|\boldsymbol{C}|\right) - \frac{1}{2}\boldsymbol{y}^{\top}\boldsymbol{C}^{-1}\boldsymbol{y} - \frac{n}{2}\log\left(2\pi\right). \tag{4.20}$$

We use the term marginal to emphasise that we are dealing with a non-parametric model. This marginal likelihood behaves somewhat differently to what one might expect from experience with parametric models. Note that it is very easy to fit the training data exactly: simply set the noise level to zero and the model produces a mean predictive function which agrees exactly with the training points. However, this is not the typical behaviour when optimizing the marginal likelihood. The log marginal likelihood, Equation 4.20 consists of three terms, the first term $-\frac{1}{2}\log\left(|\boldsymbol{C}|\right)$ acts as a complexity penalty term which measures and penalises the complexity of the model. The second term, a negative quadratic, acts as a data-fit measure (it is the only term that depends on the training set output values $\boldsymbol{y}$). The third term is a log normalisation term and is independent of the data. The tradeoff between penalty and data-fit in the Gaussian process model is therefore automatic. There is no weighting parameter which needs to be set by some external method such as cross validation.

The resulting log marginal likelihood scores for the data were standardised to the training data length. This involved dividing the log marginal likelihood score by the number of training data points per track, giving the average log marginal likelihood per point. The results are shown in Figure 4.7 and summary statistics given in Table 4.1.

	Matérn$\frac{3}{2}$	Matérn$\frac{1}{2}$	SE	NCVM
25th Percentile	0.2740	0.1427	0.2646	0.2235
Median	0.2819	0.1916	0.2794	0.2434
75th Percentile	0.3069	0.2292	0.2956	0.2530

Table 4.1: Table of log marginal likelihood scores for the Matérn$\frac{3}{2}$, Matérn$\frac{1}{2}$, squared exponential and near constant velocity kernels.

The results suggest almost comparable performance, in terms of goodness of fit, of all kernel functions. However, as shown in Figure 4.7, there is a substantial difference in the robustness. The Matérn$\frac{1}{2}$ kernel frequently finds poorer fits to the data. The squared exponential kernel performs in the middle range, occasionally finding worse solutions than the Matérn$\frac{3}{2}$ kernel but better than the Matérn$\frac{1}{2}$ kernel. The near constant velocity model kernel performs somewhere in the middle, occasionally finding worse and better solutions than the squared exponential kernel. These results in addition to the consideration that any implemented algorithm should operate in a timely fashion indicate that the Matérn$\frac{3}{2}$ kernel is optimal for modelling this data, due to its robust performance and requiring fewer kernel hyperparameters than the near constant velocity model. Naïvely we would expect the constant velocity model to do best out of all the kernels. However, the constant-velocity model does not take into account any target accelerations and subsequently the kernel performs poorly as it does not take these accelerations into account. This tells us that the data cannot be assumed to follow a continuous velocity.

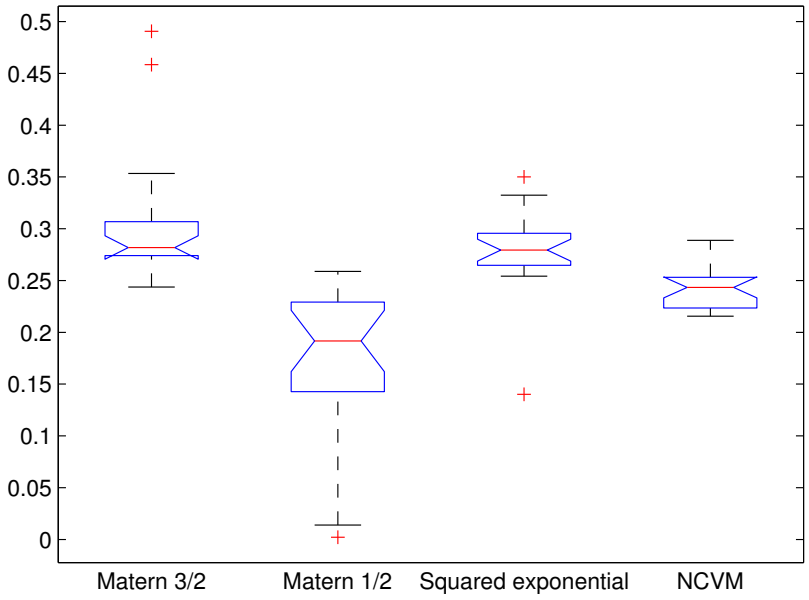


Figure 4.7: Log marginal likelihood scores for the different kernel functions applied to each track.

4.5 Summary

This chapter has developed several new Gaussian process kernels and discussed their usefulness for data modelling. The empirical results suggest that the Matérn $\frac{3}{2}$ kernel is the optimal choice for modelling maritime data. The Gaussian process now needs to be adapted in order to detect temporal anomalies in an online setting, this is next considered in Chapter 5.

Chapter 5

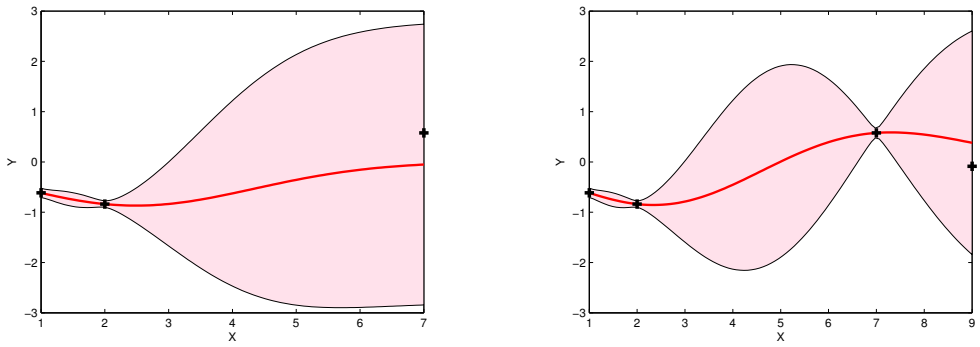
Detecting Anomalous Vessel Dynamics

Chapter 2 noted that the two approaches we entertain toward handling anomalies within data are either to accommodate them in the model, or rejection based on a discordancy metric. This chapter investigates a principled method of rejecting anomalies using Gaussian processes to model streaming data. The applicability of the developed techniques are then demonstrated through application to maritime traffic data; providing a means of identifying unusual movements or dynamics.
This chapter begins by discussing how the Gaussian process can be used to predict a mean, and uncertainty around that mean, for a given input value. This provides a means of testing for anomalies by determining whether a test data point falls within the bounds of predicted uncertainty. However, the accuracy of any anomaly detection method is dependent on the implemented methodology capturing the inherent variability of the data. The limitation of the Gaussian distribution as noted in Chapter 2 is that data is poorly modelled in the tails. Hence extreme values fail to be accounted for in a principled manner. In section 5.2.1 we discuss how, through combination with extreme value theory, a principled anomaly detection mechanism is achieved. In section 5.3 we then discusses how the approach needs to be modified, to be made amenable for online operation. Finally, the approach is applied to both synthetic (section 5.4) and real world data (section 5.5) as well as being compared against a commonly applied

method within this domain (section 5.6).

5.1 The Gaussian Process Regression Mechanism

The Gaussian process regression prediction mechanism can be used to predict for a given input value. This provides a mean value, in addition to the uncertainty in the predicted value. New data points can be included in subsequent modelling if they fall within the predicted uncertainty of the model predicted mean, an illustration of the concept is given in Figure 5.1.



(a) (b)

Figure 5.1: The Gaussian process regression mechanism predicts forward to the new data point (Figure 5.1a), the data point falls within the predictive uncertainty and is included in the subsequent update (Figure 5.1b). If the data point falls outside the predictive uncertainty then it will be excluded from subsequent updates.

As exemplified in Chapter 4, the accuracy of modelling data using the Gaussian process (and hence such an anomaly detection mechanism) is dependent on the suitability of the kernel to the data. The previous investigations indicated that the Matérn$\frac{3}{2}$ kernel is the kernel best suited to our data converted to the previously described feature space. However, this is not the only factor which determines the accuracy of the anomaly detection. The resulting probability distribution of possible output values given by the Gaussian process is itself a

Gaussian distribution. This distribution fails to accurately account for extreme values within its tails. With this in mind, we next discuss extreme value theory which can be used to model extreme observations. This leads into a discussion of how extreme value theory can then be combined with Gaussian processes in order to provide a principled approach to anomaly detection.

5.2 Extreme Value Theory

Extreme value theory has previously been used to create a novelty detection threshold, Roberts (2000); Lee and Roberts (2008), beyond which a value can be quantified as having not arisen from the underlying distribution. The theory itself focuses on the statistical behaviour of $M_n = \max\{x_1, \ldots, x_n\}$ where $x_1, \ldots, x_n$ is a sequence of independent random variables with a distribution function F. In theory the distribution of M_n can be derived exactly for all values of n, i.e.

$$\begin{aligned} Pr\{M_n \leq z\} &= Pr\{x_1 \leq z, \ldots, x_n \leq z\}, \\ &= Pr\{x_1 \leq z\} \times \ldots \times Pr\{x_n \leq z\}, \\ &= \{F(z)\}^n. \end{aligned} \tag{5.1}$$

Extreme value theory states that the entire range of possible limit distributions for M_n is given by one of three types of cumulative distribution function, types I, II and III, known as the Gumbel, Fréchet and Weibull, respectively. The Gumbel distribution is given as,

$$I : G(z) = \exp\left\{-\exp\left[-\left(\frac{z-\beta}{\alpha}\right)\right]\right\} \qquad -\infty < z < \infty. \tag{5.2}$$

Each family has a scale and location parameter, α and β respectively. Additionally the Fréchet and Weibull families have a shape parameter ξ Coles (2001). The underlying target distribution, F, in our case is taken to be Gaussian, as it is the posterior of a Gaussian process. When F is Gaussian the extreme value

distribution is of Gumbel form Coles (2001).
Assuming that some "normal" data is identically and independently Gaussian distributed, the extreme quantiles can be obtained by inverting Equation 5.2, such that for some $0 \leq p \leq 1$,

$$z_p = \beta - \alpha \log\left(-\log\left(p\right)\right). \tag{5.3}$$

The value of p which in this work is set to 0.95, hence induces a novelty threshold z_p, above which a test point is classified "abnormal". The parameters α and β require estimation and typically depend on the sample size n of the data set. These forms are only asymptotically correct as $n \Rightarrow \infty$, however as shown in Roberts (2000) estimation in closed form provides a good fit between the closed form solution and experimental data. The decoupled estimators are used for α and β, given respectively as,

$$\alpha = \left(2\log\left(n\right)\right)^{-\frac{1}{2}}, \tag{5.4}$$

$$\beta = \left(2\log\left(n\right)\right)^{\frac{1}{2}} - \frac{\log\left(\log\left(n\right) + \log\left(4\pi\right)\right)}{2\left(2\log\left(n\right)\right)^{\frac{1}{2}}}. \tag{5.5}$$

Extreme value theory therefore provides a principled means of detecting anomalies (as opposed to an ad-hoc approach), which we can use to model the extreme values within the underlying distribution. In the next section we consider how the Gaussian process can be coupled with extreme value theory.

5.2.1 Gaussian Process-Extreme Value Theory (GP-EVT)

Gaussian processes and extreme value theory have been used together within the novelty detection framework for some time Clifton et al. (2013). However, much of the existing work on novelty detection using extreme value methods has focused on non-sequential conditions; or more precisely, on a fixed training data set. Whilst the extreme value estimated from a fixed sample size will adequately account for the changes in belief about the location of extreme events, the framework is rarely extended to account for dynamic changes in the underlying generating distribution and changes in the sample size.

In this thesis system dynamics are modelled using a Gaussian process regression mechanism. At some arbitrary point in the future, $\boldsymbol{x}_*$, the Gaussian process can be interrogated and the predictive (Gaussian) distribution at that point calculated, conditional on the trajectory's past samples. This predictive distribution, which now features a context (time) dependent mean, $\boldsymbol{m}_*$, and variance, $\boldsymbol{v}_*$, allows rescaling of the extreme event quantile $\boldsymbol{e}$ (the value taken for a given probability from the cumulative distribution function Equation 5.2) such that,

$$\boldsymbol{e} = \boldsymbol{m}_* + \sqrt{\boldsymbol{v}_* \boldsymbol{z}_p}, \tag{5.6}$$

so reflecting temporal changes in the statistics of the base distribution.
In order to infer a proxy for the number of data points $n(\boldsymbol{x}_*)$ at $\boldsymbol{x}_*$ in Equation 5.11, a Gaussian kernel smoother is applied such that the estimated $n(\boldsymbol{x}_*)$ at $\boldsymbol{x}_*$ is given by,

$$n(\boldsymbol{x}_*) = \sum_{i}^{n} \phi_h(\boldsymbol{x}_*; \boldsymbol{x}_i), \tag{5.7}$$

where $\phi_h(\boldsymbol{x}_*; \boldsymbol{x}_i)$ is a (non-normalised Gaussian) radial basis function,

$$\phi_h(\boldsymbol{x}_*; \boldsymbol{x}_i) = \exp\left\{-\frac{|\,\boldsymbol{x}_* - \boldsymbol{x}_i\,|^2}{2h^2}\right\}, \tag{5.8}$$

and $\boldsymbol{x}_i$ is the most recent observation and $|\cdot|$ denotes Euclidean distance. The kernel width h is set to be equal to twice the length scale λ in Equation 3.22. The justification being that the length scale determines the distance required between data points to cause a significant change in the underlying process. The kernel width performs the same operation, considering the distance at which data points should contribute more or less significantly to the estimated data density. By setting the kernel width as equal to twice the length scale (to account for the symmetry of the radial basis function) a consistent belief for the inferred underlying process is propagated. This coupling of λ to the Gaussian process regression model ensures that tracks with long correlation lengths and smaller sampling rates will feature the same sensitivity to outliers as tracks with short correlation lengths and high sampling rates. Also, the coupling ensures that the smoothing of the sampling process does not come at a cost of an additional

parameter which would require estimation or ad hoc choice.
With an estimate for the expected number of observations (via Equation 5.7), the extreme value distribution parameters can be updated to reflect the dynamics of the sampling process. Thus, the scaling (Equation 5.4), and location (Equation 5.5), parameters can be estimated, using the proxy number of data points, n, contributing information at the location of interest $\boldsymbol{x}_*$ Miller et al. (1992).
The Gaussian process now provides a mechanism to predict the distribution of future values and to adjust the scaling of the extreme value quantile. The kernel smoothing approach applied to the sampling process provides an estimate of the future sample size. The combination of predictive distribution and sample size estimate enables sequential re-evaluation of the extreme value parameters and a subsequent novelty threshold. If the new data point value falls within some range of the predicted value then the new data point is included in the model update. The key advantage of this approach lies in the incorporation of future uncertainty in both sampling and observation processes to provide the means for a more accurate novelty detection algorithm. The graphical model representation of the complete model is shown in Figure 5.2.

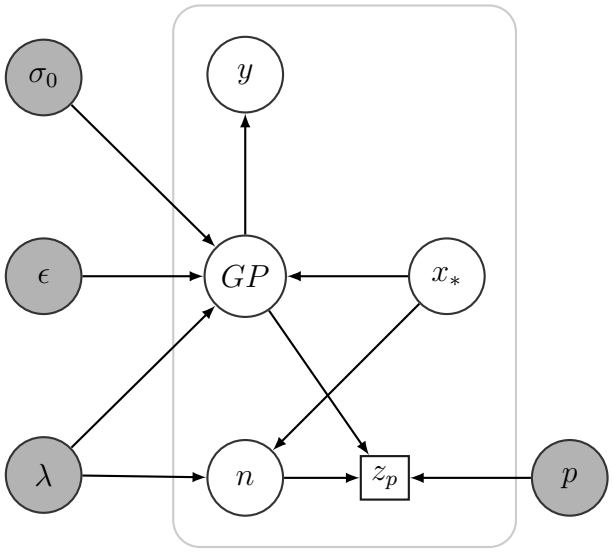


Figure 5.2: A graphical model representation of the GP-EVT model. At the centre is the Gaussian process which models the track's dynamics. It has fixed (but pre-inferred) hyper-parameters shown as grey nodes to the left. Also shown is the estimate of the sample size, n. The extreme value percentile is a deterministic node, shown as a square box, which depends upon p, the novelty level, and sample size n.

As can be seen in Figure 5.2 the model is reliant on a number of hyperparameter values. In a fully Bayesian treatment the posterior distribution for each of the hyperparameter values would need to be evaluated. However, in general, exact marginalisation will be intractable and we must resort to approximations. The simplest approach (which is undertaken in this thesis) is to make a point estimate of the hyperparameters by maximising the log marginal likelihood, Equation 4.20. This section has described a means whereby Gaussian processes and extreme value theory can be combined to provide a principled mechanism for anomaly detection. However, up until now we have not considered the issue with using Gaussian processes for online analysis. We address this issue in the next section and illustrate how the Gaussian process can be adapted for online operation.

5.3 Sequential Gaussian Process Updates

In any practical problems data is received sequentially and the total data set can grow to arbitrarily large size. It can be noted by considering the Gaussian process equations,

$$\begin{aligned} p\left(\boldsymbol{y}_*\right) &= \mathcal{N}\left(\boldsymbol{y}_*; \boldsymbol{m}\left(\boldsymbol{x}_*\right), \boldsymbol{v}\left(\boldsymbol{x}_*\right)\right), \\ \boldsymbol{m}\left(\boldsymbol{x}_*\right) &= \boldsymbol{k}^\top \boldsymbol{C}^{-1} \boldsymbol{y}, \\ \boldsymbol{v}\left(\boldsymbol{x}_*\right) &= \boldsymbol{c} - \boldsymbol{k}^\top \boldsymbol{C}^{-1} \boldsymbol{k}, \end{aligned} \tag{5.9}$$

that if the Gaussian process mechanism was continually updated in the light of new observations, then the inversion of the covariance matrix, $\boldsymbol{C}$, would be repeated with every new observation. This inversion is expensive as its computational complexity grows as $O(n^3)$ in the number of samples, i.e. the dimension of the covariance matrix, $\boldsymbol{C}$. Closer inspection however, reveals that covariance matrix $\boldsymbol{C}$, is changed on observation of a simple data point only in the addition of one new row and column. It is thus possible to reformulate the matrix inversion process as a sequential Cholesky decomposition Osborne (2010).
By decomposing a matrix into the product of a lower triangular matrix, $\boldsymbol{R}$, and its conjugate transpose the covariance matrix can be expressed as,

$$\boldsymbol{C}\left(\boldsymbol{x}, \boldsymbol{x}\right) = \boldsymbol{R}\left(\boldsymbol{x}, \boldsymbol{x}\right)^\top \boldsymbol{R}\left(\boldsymbol{x}, \boldsymbol{x}\right). \tag{5.10}$$

Based on this decomposition, the predictive distribution can be expressed by a mean and variance,

$$\begin{aligned} \boldsymbol{m}_*\left(\boldsymbol{x}_*\right) &= \boldsymbol{m}\left(\boldsymbol{x}_*\right) + \boldsymbol{b}_{\boldsymbol{x},\boldsymbol{x}_*}^\top \boldsymbol{a}_{\boldsymbol{x}} \boldsymbol{C}\left(\boldsymbol{x}_*, \boldsymbol{x}_*\right), \\ \boldsymbol{v}_*\left(\boldsymbol{x}_*\right) &= \boldsymbol{C}\left(\boldsymbol{x}_*, \boldsymbol{x}_*\right) - \boldsymbol{b}_{\boldsymbol{x},\boldsymbol{x}_*}^\top \boldsymbol{b}_{\boldsymbol{x},\boldsymbol{x}_*}, \end{aligned} \tag{5.11}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are given as,

$$\begin{aligned} \boldsymbol{a}_{\boldsymbol{x}} &\triangleq \boldsymbol{R}\left(\boldsymbol{x},\boldsymbol{x}\right)^{\top} \backslash \left(\boldsymbol{y}-\boldsymbol{m}\left(\boldsymbol{x}\right)\right), \\ \boldsymbol{b}_{\boldsymbol{x},\boldsymbol{x}_*} &\triangleq \boldsymbol{R}\left(\boldsymbol{x},\boldsymbol{x}\right)^{\top} \backslash \boldsymbol{C}\left(\boldsymbol{x},\boldsymbol{x}_*\right). \end{aligned} \tag{5.12}$$

It should be noted that the symbol $\backslash$ is used to indicate matrix division e.g. $\boldsymbol{A}\backslash\boldsymbol{B}$ indicating $\boldsymbol{A}$ divided into $\boldsymbol{B}$. This therefore performs the same operation as $\boldsymbol{A}^{-1} \times \boldsymbol{B}$ but avoids the computational demands of inverting the matrix.
When new data x_n is observed, the covariance matrix $\boldsymbol{C}$ is changed only in the addition of some new rows and columns, i.e.

$$\boldsymbol{C}(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n}) = \begin{pmatrix} \boldsymbol{C}\left(\boldsymbol{x}_{1:n-1},\boldsymbol{x}_{1:n-1}\right) & \boldsymbol{C}\left(\boldsymbol{x}_{1:n-1},\boldsymbol{x}_n\right) \\ \boldsymbol{C}\left(\boldsymbol{x}_n,\boldsymbol{x}_{1:n-1}\right) & \boldsymbol{C}\left(\boldsymbol{x}_n,\boldsymbol{x}_n\right) \end{pmatrix}. \tag{5.13}$$

Consequently the Cholesky decomposition can also be computed iteratively Osborne (2010) as,

$$\boldsymbol{R}\left(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n}\right) = \begin{pmatrix} \boldsymbol{R}\left(\boldsymbol{x}_{1:n-1},\boldsymbol{x}_{1:n-1}\right) & \boldsymbol{S} \\ \boldsymbol{0} & \boldsymbol{U} \end{pmatrix}, \tag{5.14}$$

where,

$$\begin{aligned} \boldsymbol{S} &= \boldsymbol{R}\left(\boldsymbol{x}_{1:n-1},\boldsymbol{x}_{1:n-1}\right)^{\top} \backslash \boldsymbol{C}\left(\boldsymbol{x}_{1:n-1},\boldsymbol{x}_{1:n}\right), \\ \boldsymbol{U} &= \mathrm{chol}\left(\boldsymbol{C}\left(\boldsymbol{x}_n,\boldsymbol{x}_n\right) - \boldsymbol{S}^{\top}\boldsymbol{S}\right). \end{aligned} \tag{5.15}$$

With this Cholesky update expressed iteratively the predictive distribution, Equation 5.11, can also be expressed iteratively by expressing the vector $\boldsymbol{a}$, in Equation 5.12, via the simple update rule,

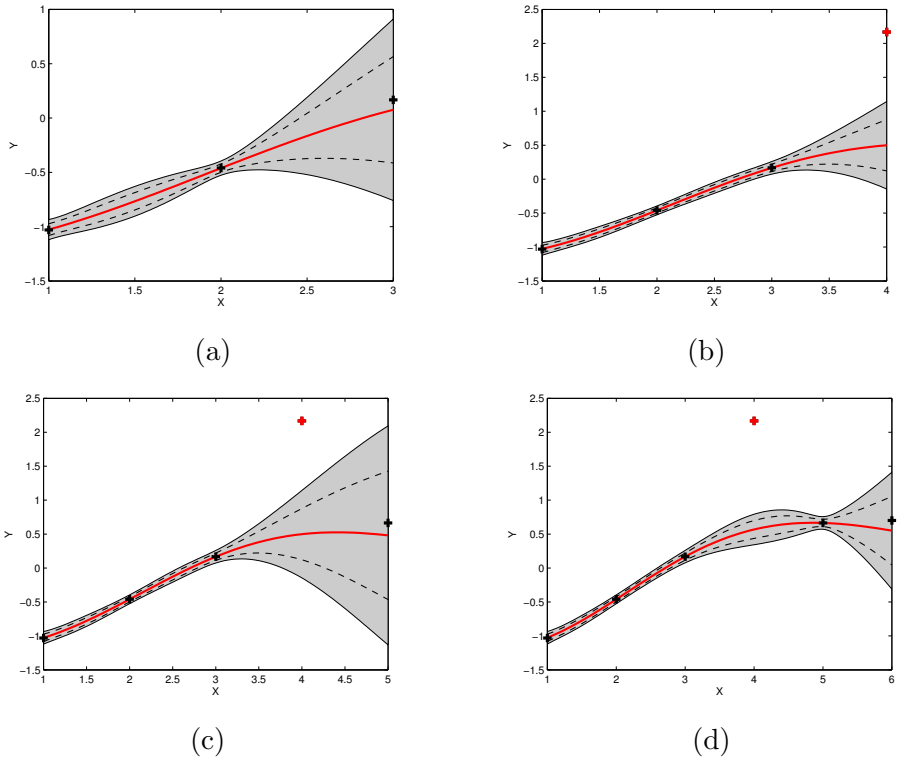
$$\boldsymbol{a}_{1:n} = \begin{pmatrix} \boldsymbol{a}_{1:n-1} \\ \boldsymbol{U}^{\top} \backslash \left(\boldsymbol{y}_n - \boldsymbol{m}\left(\boldsymbol{x}_n\right) - \boldsymbol{S}^{\top}\boldsymbol{a}_{1:n-1}\right) \end{pmatrix}. \tag{5.16}$$

This avoids the computationally expensive matrix inversion, in Equation 3.20, and allows the Cholesky factor to be expressed as an efficient update rule.
The efficacy of the approach presented in this and previous sections will next be

demonstrated through application to both synthetic and real data. In the next section synthetic data will first be used to illustrate some of the features of the method, the section following this then demonstrates the methods applicability through application to real vessel track data.

5.4 Synthetic Data Illustration

To enable a simple exposition of the method, synthetic data was generated from a Gaussian process with a Matérn$\frac{3}{2}$ kernel, (Equation 3.22), with parameters set to $\sigma_0 = 1$, $\lambda = 2$ and $\sigma = 0.01$. Anomalies were generated by offsetting randomly selected samples that were previously drawn from the Gaussian process by a fixed offset value, making the data point anomalous with respect to surrounding points. Hyperparameters were inferred from an anomaly free portion of the data set using a point estimate of their true values (Equation 4.20). In the case of the kernel width, h, used to estimate the density of data points (Equation 5.8), this was set as equal to twice the inferred length scale hyperparameter value. This fixed kernel width was used in order to estimate the number of data points n, at each x_*. The Gaussian process extreme value theory mechanism applied to this data set is illustrated in Figure 5.3.



(a) (b) (c) (d)

Figure 5.3: In the above illustration of the Gaussian process extreme value theory anomaly detection mechanism the continuous red line shows the predicted mean function, and the grey areas show the extreme value theory bound of the Gaussian process predictive distribution for $p = 0.95$. The dashed line shows the error bound produced by considering the 95% bound from the mean function (1.64 standard deviations from the mean). In Figure 5.3a the Gaussian process predicts forward to a new data point. The grey bound is open to the right and widening until the next observation has been included. In Figure 5.3b we can see that GP-EVT has classified the observation as non-anomalous, as it has been included within the update. The Gaussian process is now also predicting forward toward the artificial anomaly at $x = 4$. This data point falls outside the error bounds and will therefore be excluded from any subsequent updates. In Figure 5.3c we can see that after detecting and excluding the artificial anomaly the uncertainty bounds continue to increase (until they reach their maximum as set by the prior distribution). The subsequent observations fall well within the error bound and so will be included in the next update as in Figure 5.3d.

It can be observed in Figure 5.3 how each new data point is considered with respect to the previously learned underlying function. If the new point falls

within the predictive uncertainty of the next data point it is included in the sequential update otherwise it will be excluded. An example of such an update step is shown in Figure 5.3a. The new data point falls outside the extreme value theory bound, Equation 5.6, and so has been excluded. Notice that if a data point has not been observed for a period, the predictive uncertainty grows, allowing for the possibility of a dynamic change in the underlying base function and the new data point to be included in the update. In this manner anomalous points within the data can be clearly identified while perfectly accommodating for the dynamics of the underlying function and the irregular nature of its observation.

In order to illustrate how the irregularity of observed data points effects the scaling of the extreme value distribution, and hence the novelty bounds, a Gaussian process predictive distribution was calculated for 1000 samples within the windowed region of track. Again a fixed kernel width was used in order to estimate n at each x_*. The window ending at the time period for the new observed sample is given in Figure 5.4 and Figure 5.5. Both plots show a snapshot from the last observed sample. In 5.4 the observation rate is high, while in 5.5 the observation rate is low. The observation density affects the location of the probability density function of the extreme value distribution $f_e(y)$ (blue lines, lower plot), relative to the predicted Gaussian probability distribution function $f(y)$ (red dashed line, lower plot), drifting closer to the base distribution as the density of points decreases.

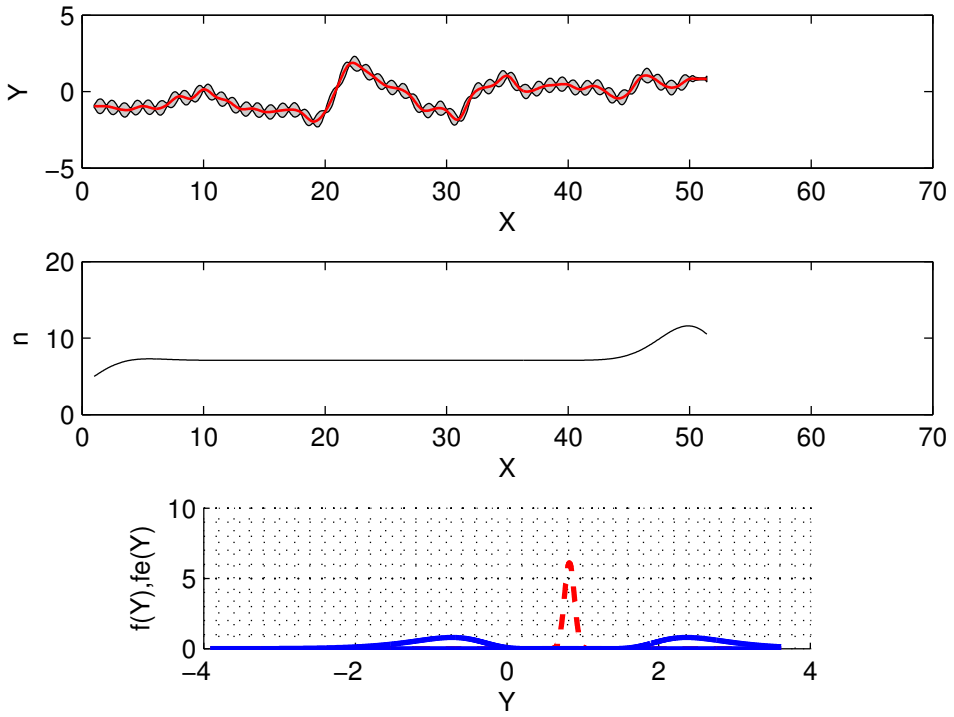


Figure 5.4: In the above illustration the upper plot shows the sequential GP-EVT mechanism stopped at a region of high observation density, as evidenced by the middle plot of observation density. The grey bounds of the GP-EVT mechanism are the result of applying Equation 5.6 to the result from the Gaussian process regression of the data and the calculation of the extreme quantile at each data index, Equation 5.3. The estimate of the number of data points in the middle plot, which is used in the calculation of the extreme quantile location and scale parameters, is produced through application of Equation 5.7. As can be observed the number of data points which contribute to the Gaussian process inference has increased significantly, as indicated by the observation density shown in the middle plot. Consequently the location of the extreme value distributions, illustrated by the continuous lines in the bottom plot, move away from the posterior predictive distribution (dashed line). The plot of the extreme value distributions is bimodal to illustrate that the extreme quantile is symmetrically applied to the mean from the Gaussian process regression.

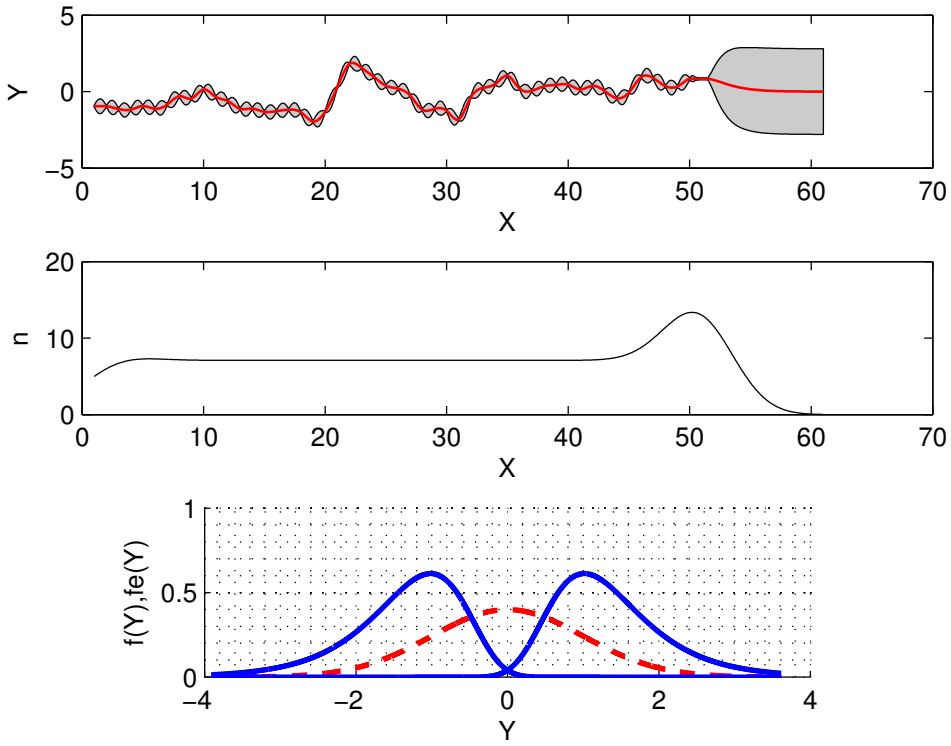


Figure 5.5: In the above illustration the top plot shows the sequential GP-EVT, stopped at a region of very low observation density, as evidenced by the middle plot of observation density. Note that this figure includes the results illustrated within Figure 5.4 as a subset of the illustrated results and so contains both periods of high and low observation density. In the lower plot the predictive distribution (dashed line) is an accurate representation of the true distribution (continuous line). This is due to the relationship between the observation density and the location and scaling of the extreme value distribution.

5.5 Vessel Track Anomaly Detection

As mentioned in the previous chapter, approximately 100 real world vessel tracks were collected through a custom written web scraping application, used to collect freely available information (the code for which is given in Appendix C). To reduce the required processing time per track and in order to detect anomalous changes in the ship dynamics (sudden changes in a vessels acceleration) the data was first

converted to an appropriate feature space (previously discussed in Chapter 4).

5.5.1 Vessel Track Modelling

In this section the GP-EVT methodology is applied to a real world vessel track. The Matérn $\frac{3}{2}$ kernel was chosen to model the underlying dynamics using hyper-parameters learnt from a subset of 30 tracks available from the total data. Each track was chosen to be sufficiently long enough so that the underlying dynamics of the vessels could be captured.

Figure 5.6 shows an example vessel track without outlying points. The track is from a dredger which follows a smooth trajectory and does not make any sudden changes in acceleration. Shown in Figure 5.7 are the sequential extreme value theory bounds sea-sawing until the next observation arrives. All observations fall well inside the predictive boundary of the GP-EVT bound and, consequently, no anomalies are detected.

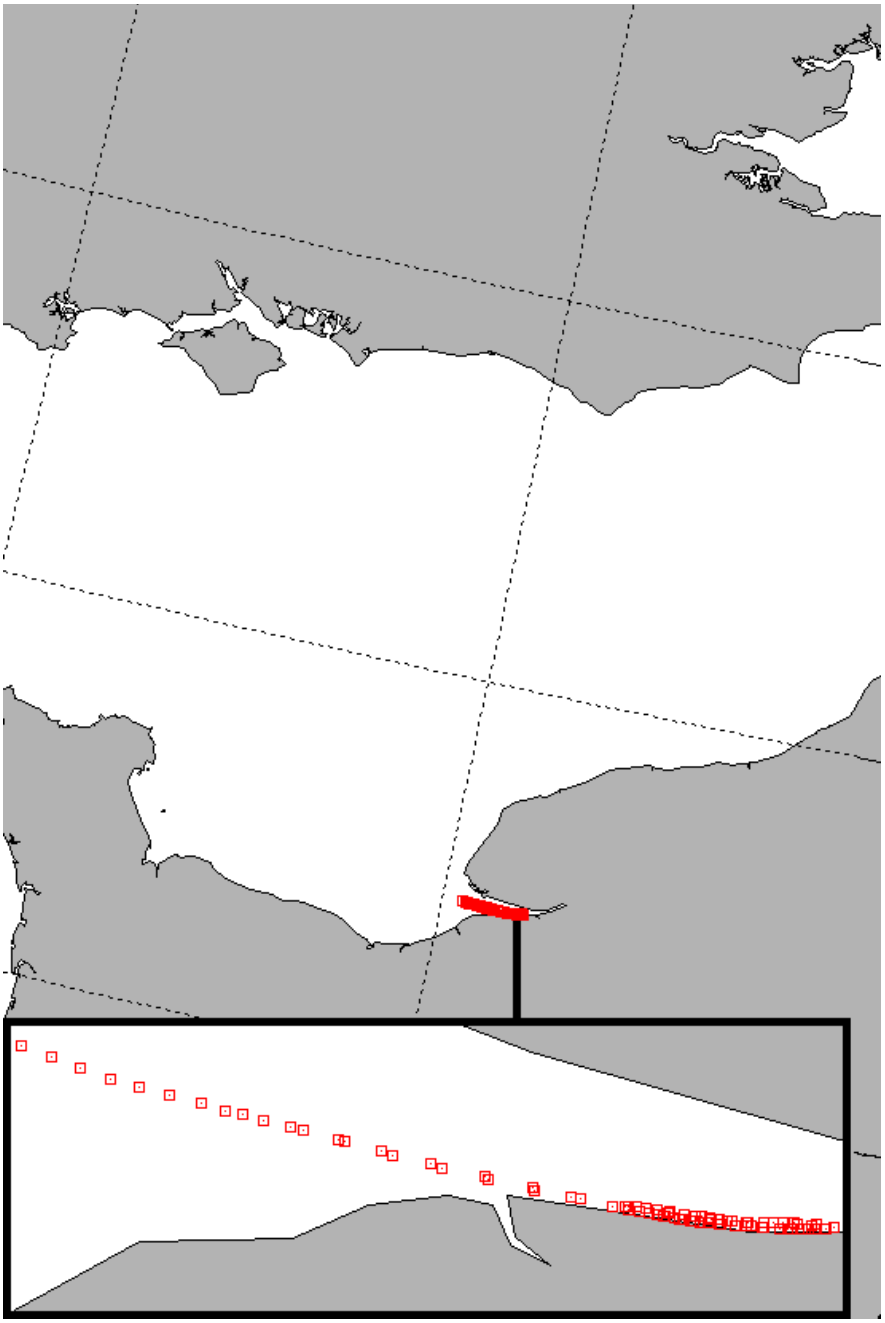


Figure 5.6: A plot of the anomaly free GPS track from a dredging vessel operating off the coast of France, near Le Havre.

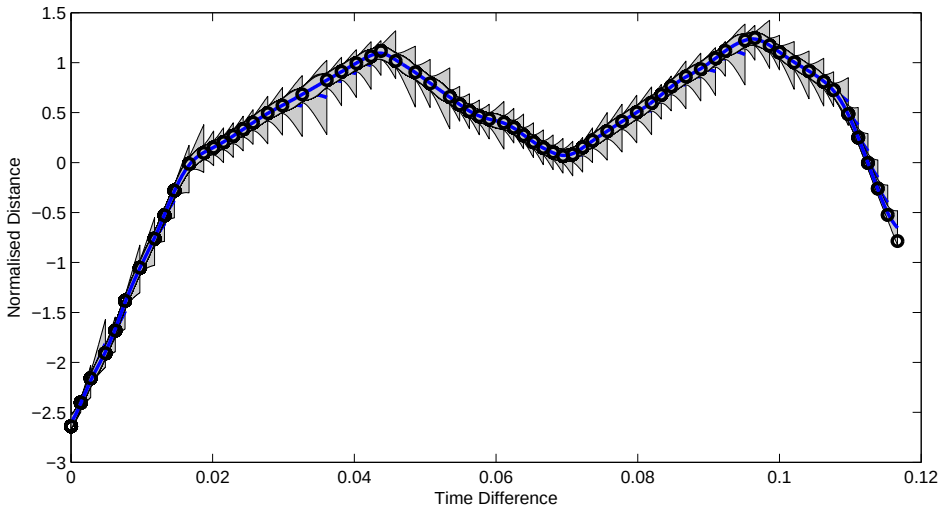


Figure 5.7: Sequential predictions applied to feature extracted data, also showing that all data points fall within the extreme value theory bound.

Figure 5.8 shows an example of a vessel track with some points which the GP-EVT model labels as anomalies. As can be seen in Figure 5.8, the vessel remains within a confined area and there are short sudden movements, Figure 5.9. These are marked as anomalies and are perhaps the result of the vessel drifting, manoeuvering or being moored. Figure 5.10 also shows an enlarged section of the graphical application the sequential GP-EVT model.

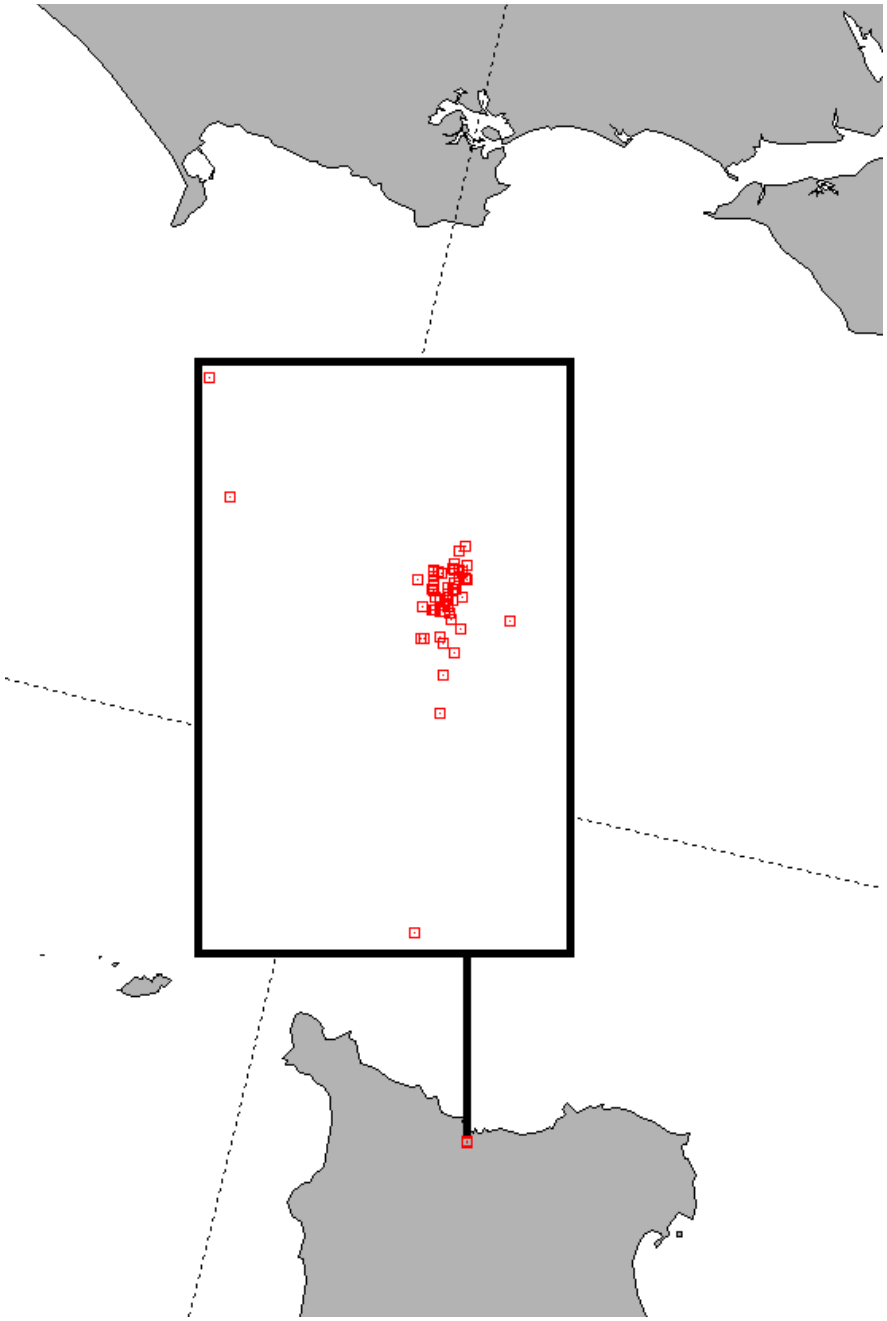


Figure 5.8: A plot of the GPS track from a small vessel operating off the coast of France near Cherbourg and whose track suggests unusual navigation behaviour.

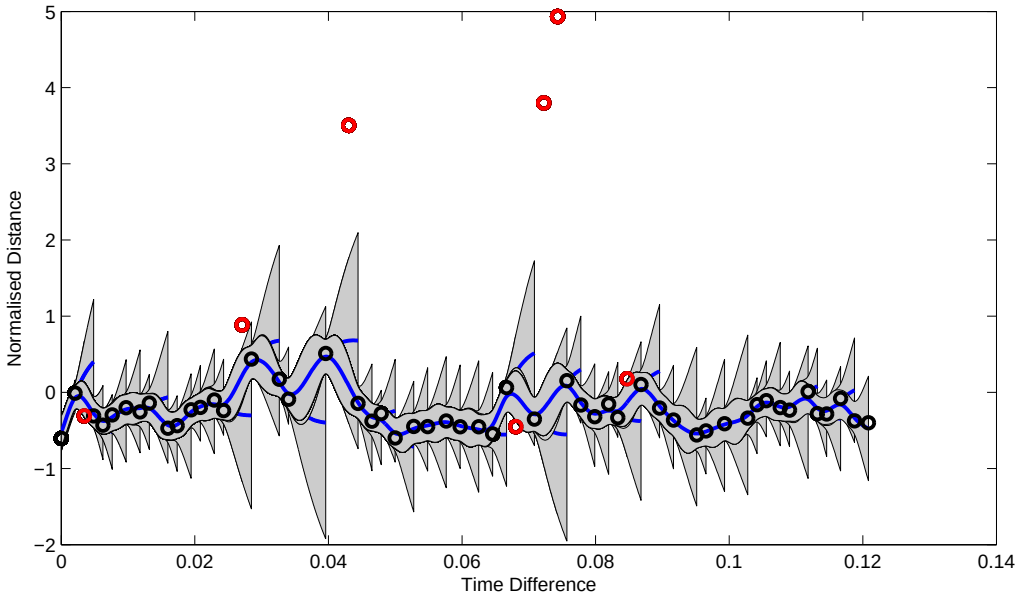


Figure 5.9: Sequential predictions applied to feature extracted data, also showing some data points that fall outside the GP-EVT bound.

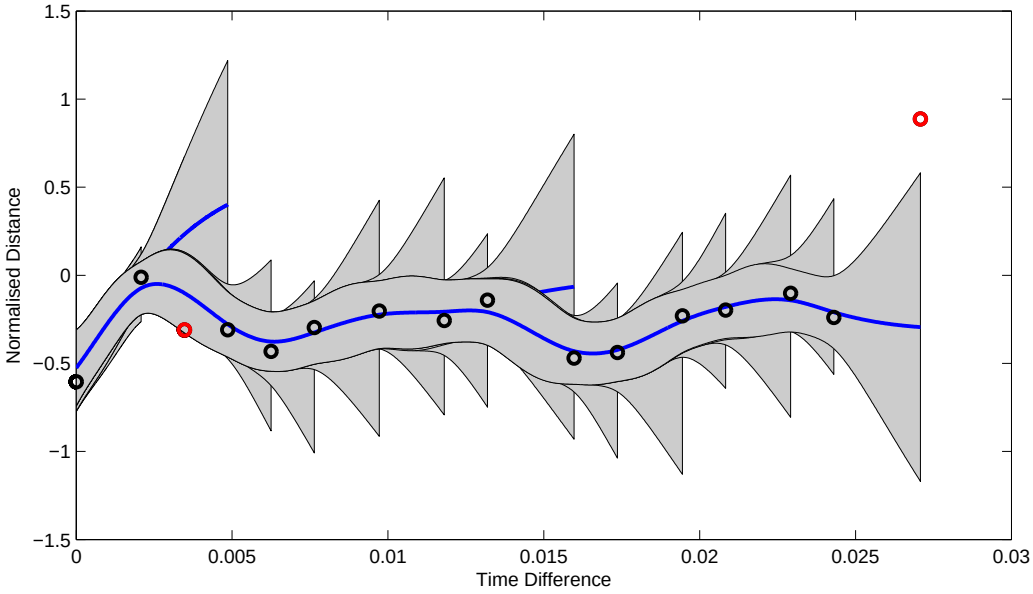


Figure 5.10: Magnified section of the plot in Figure 5.9. The GP-EVT bound has predicted forward to the new data point, as the data point falls outside of the bounds of the GP-EVT model it will be excluded from the model update. The grey shaded areas indicate the extreme value theory bound scaled by the Gaussian process uncertainty.

5.6 Qualification and Comparison

In this section a traditional approach to anomaly detection using the Kalman filter is compared with the GP-EVT approach, in order to validate the improvement offered by coupling Gaussian processes with extreme value theory. The traditional approach uses a Kalman filter to model the normal behaviour of the ship Laws et al. (2011). Data is then regarded as anomalous if it is more than a fixed number of standard deviations from the mean Manson et al. (2000) (typically 3 to 5 standard deviations) Wang et al. (2011).

To obtain a data set for comparison of the methodologies a subset of the total collected data (14 tracks) for a variety of vessel types was selected. This data was allocated a human expert operator who studied the feature space representation of the data. Data points believed to exhibit an unusual change in the vessel

dynamics given its previous course were then labeled as anomalous. Individual model parameters for both the Kalman filter parameters and Gaussian process model hyperparameters were then inferred from a further subset of the total data consisting of 30 tracks. The mean set of inferred model parameters was then used in each model.

To ensure a valid comparison between the alternate approaches, we use the area under the ROC curve as this offers a more robust metric Hamel (2008). Comparisons were made using four anomaly thresholds, designed such that they represented the same percentiles on the cumulative distributions of either a Gaussian (for the non-EVT approach) or the EVT distribution. These four thresholds were chosen to be 1, 1.64, 3 and 5 standard deviations for the non-EVT approach and 0.84, 0.95, 0.99 and 0.999 for the models using EVT.

The Kalman filter approach requires a process model of the normal behaviour of the ship. Typically, a near constant velocity model is chosen to model the continuous trajectory without imposing any excessive smoothness on the trajectory George et al. (2011). It has previously been shown in Chapter 4 that the Matérn$\frac{3}{2}$ is the most suitable Gaussian process kernel for modelling vessel track data. To provide fair comparison between the GP-EVT method using the Matérn$\frac{3}{2}$ kernel and the Kalman filter, the Kalman filter must be implemented with a second order differentiable systems model. Using the Kalman filter with the near constant velocity model therefore provides a fair comparison, as both Matérn$\frac{3}{2}$ and constant velocity model are second order differentiable. To thoroughly validate the GP-EVT approach a traditional Kalman filter using a number of fixed standard deviations to exclude anomalies and a Kalman filter using the extreme value theory in a manner similar to the Gaussian process has also been compared. By so doing, a comparison has been made to both models of normal ship behaviour (namely the Matérn$\frac{3}{2}$ and the near constant velocity model) and also both approaches to detecting and excluding anomalies (namely, the extreme value theory and standard deviation approaches).

When using the Kalman filter the mean and standard deviation were predicted forward to the same time step as the new observation. If the point lies within a pre-chosen confidence region (defined as a multiple of the standard deviation about the mean) it is included in the update. Thus we make one classification per

data point either normal or abnormal. This was repeated for a range of confidence regions defined by different multiples of the standard deviation. The resulting area under curve for the receiver operating characteristics, which compares the Kalman filter using the near constant velocity model (with and without extreme value theory) against the Gaussian process using a Matérn$\frac{3}{2}$ model (again with and without the extreme value theory) are shown in Table 5.1. The area under the curve takes into account all the data sets and allows for a fair comparison between methods. It can be noted that both the Kalman filter and Gaussian process performance is significantly improved using the extreme value theory. This is due to the fact that the standard deviation approach uses a fixed threshold which does not take into account the density of observations i.e. as more sample are observed a better understanding of the true distribution of values is gained. The extreme value theory, however, uses a dynamic threshold which takes into account the density of observations therefore better utilises available information to adjust the threshold.

GP-EVT	GP	KF-EVT	KF
0.8032	0.7889	0.6545	0.6119

Table 5.1: Area under curve for the Kalman filter using the near constant velocity model (with and without extreme value theory) and Gaussian process using a Matérn$\frac{3}{2}$ model (again with and without the extreme value theory) applied to 14 labelled tracks.

Although the results indicate a significant improvement of the GP-EVT over the Kalman filter approach, this is a limitation of the near constant velocity model used and not a critique of Kalman filter based methods. It can be noted that the Matérn$\frac{3}{2}$ Gaussian process model can be efficiently implemented within the Kalman filter as a Markov process model Hartikainen and Särkkä (2010). Thus, it is possible to match the area under the curve of the Kalman filter approach and Gaussian process approach by replacing the near constant velocity model in the

Kalman filter by the Markovianised Matérn model as described in Hartikainen and Särkkä (2010).
The results do however illustrate the significant improvement obtained by combining such methods with extreme value theory as opposed to a fixed number of standard deviations between a data point and the expected position of the ship.

5.7 Summary

This chapter has explored the combination of extreme value statistics with Gaussian processes, with applicability to the detection of single anomalous data points from a vessel track. Sequential updates of the Gaussian process probability distribution are made governing these extreme values, enabling context sensitive decisions. The technique has then been applied to a sample data set for comparison against a typical method of anomaly detection using Kalman filters, the results illustrate significant improvement through combination with extreme value theory. However, the technique only enables us to detect single anomalous observations it does not consider the vessel track as a whole. In the next chapter we therefore extend the anomaly detection to consider the entire vessel track, which we use to address the issue of vessel class mis-representation.

Chapter 6

Identifying Anomalous Vessel Tracks

This chapter exploits techniques from the field of divergence measurement through application of the Hellinger distance, providing a means of measuring the similarity between different vessel tracks. Vessel tracks are first modelled using a multiple output Gaussian process regression, in order to capture vessel dynamics. Bayesian non-negative matrix factorization is then applied as means of identifying overlapping community (clusters of functional data) structure. The applied methods provide a quantifiable means of expressing the degree of community membership each track is believed to hold. This addresses an aspect of marine situational awareness, which is concerned with comprehending the maritime environment, specifically of Automatic Identification System (AIS) data. AIS is an automatic tracking system based on reports provided by the vessels carrying an AIS transponder. The reports contain information on the vessel position, velocity, vessel class etc. and typically have high accuracy. Given that AIS is a self-reporting system, the trustworthiness of data depends on the data being reported by a vessel, rather than measured by a sensor. AIS data is therefore prone to be exploited, through spoofing of signals, for the benefit of illegal operations. Through detecting communities of functional data types (shipping vessels) it is possible to identify one possible type of spoofing (vessel type misrepresentation), by identifying anomalies between the broadcast vessel type and the community

to which it is assigned.

Methods of community detection provide a means of grouping together graph nodes exhibiting similar characteristics Newman (2010). Many of these methods detect communities through optimization of the Newman modularity, which can give poor results if the network is sufficiently large Fortunato and Barthélemy (2007). The Newman Modularity is a popular measure for the structure of networks, designed to measure the strength of division of a network into communities. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. Bayesian non-negative matrix factorization (NMF) does not suffer the drawbacks of modularity optimization methods, such as the resolution limit, furthermore it provides a means of soft partitioning (node member overlap) Psorakis et al. (2011).

Detection of community structure is reliant on the creation of a matrix of distances (an adjacency matrix); in this instance this is not simply a distance in the spatial sense but must express the distance in the intrinsic dynamics that created the track; i.e. the track shape. Many methods of defining distance between probability distribution exist Basseville (1989). However, only true metrics produce a symmetric adjacency matrix Budzyński et al. (2008) and such a metric is considered.

6.1 Hellinger Distance

Identification of community structure is based on some measure of adjacency or distance between functions. The concept of distance between two probability distributions is used to reflect that some probability distributions are "closer together" and consequently that it may be easier to distinguish between a pair of distributions which are "far from each other" than between those which are closer. In terms of community identification, functions exhibiting similar characteristics will be much closer in distance than those generated from functions with significantly different characteristics. Due to the uncertainty surrounding the network structure the distance between any two tracks must be considered as being symmetric, in order to avoid bias. Therefore we wish to consider only true metrics which will produce a symmetric adjacency matrix. The Hellinger

distance is one such measure of similarity between two probability distributions and is defined,

$$h^2\left(f\left(\boldsymbol{x}\right),g\left(\boldsymbol{x}\right)\right)=\frac{1}{2}\int\left(\sqrt{f\left(\boldsymbol{x}\right)}-\sqrt{g\left(\boldsymbol{x}\right)}\right)^2 d\boldsymbol{x}, \tag{6.1}$$

where $f\left(\boldsymbol{x}\right)$ and $g\left(\boldsymbol{x}\right)$ denote probability distributions of the same form. Equation 6.1 can also be alternatively expressed as,

$$\begin{aligned}h^2\left(f\left(\boldsymbol{x}\right),g\left(\boldsymbol{x}\right)\right)=&\frac{1}{2}\int\left(\sqrt{f\left(\boldsymbol{x}\right)}\sqrt{f\left(\boldsymbol{x}\right)}-\sqrt{f\left(\boldsymbol{x}\right)}\sqrt{g\left(\boldsymbol{x}\right)}-\sqrt{f\left(\boldsymbol{x}\right)}\sqrt{g\left(\boldsymbol{x}\right)}\right.\\&\left.+\sqrt{g\left(\boldsymbol{x}\right)}\sqrt{g\left(\boldsymbol{x}\right)}\right)d\boldsymbol{x},\\=&\frac{1}{2}\int\left(g\left(\boldsymbol{x}\right)+f\left(\boldsymbol{x}\right)-2\left(\sqrt{f\left(\boldsymbol{x}\right)}\sqrt{g\left(\boldsymbol{x}\right)}\right)\right)d\boldsymbol{x},\\=&1-\int\left(\sqrt{f\left(\boldsymbol{x}\right)}\sqrt{g\left(\boldsymbol{x}\right)}\right)d\boldsymbol{x}.\end{aligned} \tag{6.2}$$

In a Gaussian process the final form is a Gaussian distribution given by a mean and covariance function. Therefore to calculate the Hellinger distance between time series modelled by Gaussian processes the distributions ($f\left(\boldsymbol{x}\right)$ and $g\left(\boldsymbol{x}\right)$) must be multivariate Gaussian. In this instance the Hellinger distance would take the following form,

$$\begin{aligned}&h^2\left(f\left(\boldsymbol{x};\boldsymbol{\mu_f},\boldsymbol{\Sigma_f}\right),g\left(\boldsymbol{x};\boldsymbol{\mu_g},\boldsymbol{\Sigma_g}\right)\right)=\\&1-\frac{1}{\left(2\pi\right)^{\frac{n}{2}}\sqrt{\left|\boldsymbol{\Sigma_f}\right|}}\frac{1}{\left(2\pi\right)^{\frac{n}{2}}\sqrt{\left|\boldsymbol{\Sigma_g}\right|}}\int\left(\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu_f}\right)^\top{\boldsymbol{\Sigma_f}}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu_f}\right)\right)^{\frac{1}{2}}\right.\\&\left.\exp\left(\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu_g}\right)^\top{\boldsymbol{\Sigma_g}}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu_g}\right)\right)^{\frac{1}{2}}\right)d\boldsymbol{x},\end{aligned} \tag{6.3}$$

This can be alternatively expressed as,

$$
\begin{aligned}
&h^2\left(f\left(\boldsymbol{x};\boldsymbol{\mu_f},\boldsymbol{\Sigma_f}\right),g\left(\boldsymbol{x};\boldsymbol{\mu_g},\boldsymbol{\Sigma_g}\right)\right)= \\
&\quad 1-\frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Sigma_f}|}}\frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Sigma_g}|}} \\
&\quad \times\int\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu_f}\right)^{\top}\boldsymbol{\Sigma_f}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu_f}\right)-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu_g}\right)^{\top}\boldsymbol{\Sigma_g}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu_g}\right)\right)^{\frac{1}{2}}d\boldsymbol{x}.
\end{aligned}
\tag{6.4}
$$

The terms inside the exponent can also be expressed in quadratic form, $(\boldsymbol{x}-\boldsymbol{\mu}^*)\boldsymbol{C}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^*)+\boldsymbol{B}$, by making the following associations,

$$
\begin{aligned}
\boldsymbol{\mu}^* &= \left(\frac{1}{2}\boldsymbol{\Sigma_f}^{-1}+\frac{1}{2}\boldsymbol{\Sigma_g}^{-1}\right)^{-1}\left(\frac{1}{2}\boldsymbol{\Sigma_f}^{-1}\boldsymbol{\mu_f}+\frac{1}{2}\boldsymbol{\Sigma_g}^{-1}\boldsymbol{\mu_g}\right), \\
\boldsymbol{C}^{-1} &= \frac{1}{2}\boldsymbol{\Sigma_f}^{-1}+\frac{1}{2}\boldsymbol{\Sigma_g}^{-1}, \\
\boldsymbol{B} &= \left(\boldsymbol{\mu_f}-\boldsymbol{\mu_g}\right)^{\top}\left(\frac{1}{2}\boldsymbol{\Sigma_g}+\frac{1}{2}\boldsymbol{\Sigma_f}\right)^{-1}\frac{1}{4}\left(\boldsymbol{\mu_f}-\boldsymbol{\mu_g}\right).
\end{aligned}
\tag{6.5}
$$

The integral can now be solved and the expression simplified to,

$$
\begin{aligned}
&h^2\left(f\left(\boldsymbol{x}\right),g\left(\boldsymbol{x}\right)\right)= \\
&\quad 1-\frac{|\frac{1}{2}\boldsymbol{\Sigma_f}^{-1}+\frac{1}{2}\boldsymbol{\Sigma_g}^{-1}|^{-\frac{1}{2}}}{|\boldsymbol{\Sigma_f}|^{\frac{1}{4}}|\boldsymbol{\Sigma_g}|^{\frac{1}{4}}}\exp\left(-\frac{1}{8}\left(\boldsymbol{\mu_f}-\boldsymbol{\mu_g}\right)^{\top}\left(\frac{1}{2}\boldsymbol{\Sigma_g}+\frac{1}{2}\boldsymbol{\Sigma_f}\right)^{-1}\left(\boldsymbol{\mu_f}-\boldsymbol{\mu_g}\right)\right).
\end{aligned}
\tag{6.6}
$$

Under the assumption that both distributions have the same zero mean, $\boldsymbol{\mu_f}=\boldsymbol{\mu_g}=0$, this can be further simplified to,

$$
h^2\left(f\left(\boldsymbol{x};\boldsymbol{\mu_f},\boldsymbol{\Sigma_f}\right),g\left(\boldsymbol{x};\boldsymbol{\mu_g},\boldsymbol{\Sigma_g}\right)\right)=1-\frac{|\frac{1}{2}\boldsymbol{\Sigma_f}^{-1}+\frac{1}{2}\boldsymbol{\Sigma_g}^{-1}|^{-\frac{1}{2}}}{|\boldsymbol{\Sigma_f}|^{\frac{1}{4}}|\boldsymbol{\Sigma_g}|^{\frac{1}{4}}}. \tag{6.7}
$$

As our Gaussian process uses a flat zero-mean function, which is fixed, so the posterior mean function is also zero. Note that this does not imply that the

posterior expectation (the predictive mean) at any point is zero, in the same way that we do not expect samples from a zero-mean normal to be zero. Of course, we can readily make these tacit assumptions because we normalise the data, without loss of generality, to have zero mean. This has the advantage that, when we evaluate our distance metric between Gaussian processes we need not consider mean functions. Furthermore, this makes our distance metric independent of any translations and locations and dependent only on the similarity between of the form of the posterior Gaussian process function in question.

We normalise each time-series, prior to any further processing, using a simple linear regression, removing the sample mean and scaling by the reciprocal of the sample standard deviation.

To avoid the inverse covariances in this form, the fraction can be multiplied through by $(|\boldsymbol{\Sigma_f}||\boldsymbol{\Sigma_g}|)^{-\frac{1}{2}}$, in addition to application of the determinant identity $|\boldsymbol{AB}| = |\boldsymbol{A}||\boldsymbol{B}|$. The result can then be expressed as,

$$h^2\left(f\left(\boldsymbol{x};\boldsymbol{\mu_f},\boldsymbol{\Sigma_f}\right),g\left(\boldsymbol{x};\boldsymbol{\mu_g},\boldsymbol{\Sigma_g}\right)\right) = 1 - \frac{\left|\frac{1}{2}\boldsymbol{\Sigma_f} + \frac{1}{2}\boldsymbol{\Sigma_g}\right|^{-\frac{1}{2}}}{|\boldsymbol{\Sigma_f}|^{-\frac{1}{4}}|\boldsymbol{\Sigma_g}|^{-\frac{1}{4}}}. \tag{6.8}$$

It can be noted, as a means of verifying the result, that by setting $\boldsymbol{\Sigma} = \sigma^2$ Equation 6.8 is consistent with the Hellinger distance between two univariate Gaussian distributions (providing $\mu_f = \mu_g = 0$),

$$1 - \left(\frac{2\sigma_f\sigma_g}{\sigma_f^2+\sigma_g^2}\right)^{\frac{1}{2}} \exp\left(-\frac{(\mu_f-\mu_g)^2}{4\left(\sigma_f^2+\sigma_g^2\right)}\right). \tag{6.9}$$

The distance metric is now dependent only on the kernel function used to model the data. In the next section we consider how we can use the adjacency matrix resulting from the application of Equation 6.8 to discover communities of vessels exhibiting similar track characteristics.

6.2 Community Detection

By considering the adjacency structure created by the Hellinger distance it is possible to discover communities; tracks closer together (in the Hellinger metric

sense) are more likely to have been generated by the same class of vessel. Non-negative matrix factorisation community detection provides several benefits over previous methods of community detection Psorakis et al. (2011). Previously the number of communities within the model was discovered by performing multiple runs for varying numbers of communities K. Selection was based on the number of communities which maximise the Newman modularity. The computational demands of performing multiple runs for varying numbers of communities are negated in Non-negative matrix factorisation community detection by appropriate selection of the prior distribution over the latent variables.

The use of inverse Hellinger distance between GPs allows us to map our data to a relational space, where each pair i, j of vessel tracks is assigned a similarity value s_{ij}. We encode all similarity pairs to a matrix $\mathbf{S}$ so that s_{ij} is the degree of coupling between the paths of vessels i and j. From such relational structure, we seek to apply an appropriate clustering scheme in order to extract classes of vessels that exhibit similar movement patterns. In Reece and Roberts (2010) tracks were clustered using GPs and in which the most likely mixture model for the data was identified, this corresponding to the cluster or community for a given type of track. The distance measure in this instance was the likelihood function. Unfortunately, this method scales poorly with the number of tracks.

By treating $\mathbf{S}$ as an adjacency matrix from a network analysis perspective, where N vessels are nodes and similarities (in the Hellinger metric sense) are link weights, we apply the idea of *community detection*, Porter et al. (2009), in order to discover groups of strongly connected nodes, so that a given vessel i has more similar paths with vessels inside a community than with the ones outside.

Towards the above goal, we employ a Bayesian non-negative matrix factorisation (NMF) scheme, Psorakis et al. (2011), which has already been successfully applied to a wide range of community detection problems, Psorakis et al. (2011), Simpson et al. (2013), Psorakis et al. (2012). In this approach, communities are treated as explanatory latent variables for the observed link weights, so that the stronger the similarity between two vessel paths the more likely it is that they belong to the same community. We extract such latent grouping via an appropriate factorisation of $\mathbf{S} \simeq \mathbf{WH}$, $\mathbf{S} \in \mathbb{R}^{N \times N}$, $\mathbf{W}, \mathbf{H}^{\mathsf{T}} \in \mathbb{R}^{N \times K}$, where both the inner rank K and the factor elements w_{ik}, h_{ki} are inferred via an appropriate Maximum a

Posteriori (MAP) scheme. We can regard the elements w_{ik} as mixing co-efficients and the h_{ki} elements forming a basis set of community structures. This model has the advantage of not only discovering overlapping communities (soft-partitioning) but also quantifying how strongly each vessel belongs to a particular class. Such result allows us to quantify how the broadcast vessel class "disagrees" with the one we infer from the path similarity matrix. It also avoids the scaling issues as posed by the method proposed in Reece and Roberts (2010).
In the next sections we demonstrate the effectiveness of the method through application to both real and synthetic data. Synthetic data is used to illustrate the principles behind the methodology and a real world example is used to demonstrate its applicability.
In order to accurately capture and model vessel tracks Gaussian processes can be extended to capture the relationship between multiple outputs for the given input dimension (Section 3.2.3). In this instance two outputs (latitude and longitude) were considered, the output spherical covariance therefore takes the form as expressed as in Equation 3.25. The predictive equations for GP regression are given in Equation 3.20, which in this instance are assumed to have a zero mean.

6.3 Choice of Kernel Function for Vessel Modelling

As previously noted the choice of kernel function is central to accurate data modelling. To again determine the optimal kernel function for the new bivariate feature space a multiple output squared exponential kernel (Equation 3.21), Matérn$\frac{3}{2}$ kernel (Equation 3.22), Matérn$\frac{1}{2}$ kernel (Equation 3.23) were investigated. Due to a desire for inference to complete in a timely manner the near constant acceleration kernel (Equation 4.15) and near constant velocity kernel (Equation 4.9) were omitted from the investigation. Training data consisting of 30 vessel tracks was again used to estimate the GP kernel function parameters, through maximisation of the log marginal likelihood of the data Rasmussen and Williams (2006), the summary statistics are given in Table 6.1.

	Matérn$\frac{3}{2}$	Matérn$\frac{1}{2}$	Squared Exponential
25th Percentile	0.2910	0.3081	0.3003
Median	0.3345	0.3328	0.3333
75th Percentile	0.3705	0.3526	0.3461

Table 6.1: Table of log marginal likelihood scores for the Matérn$\frac{3}{2}$, Matérn$\frac{1}{2}$ and squared exponential kernels.

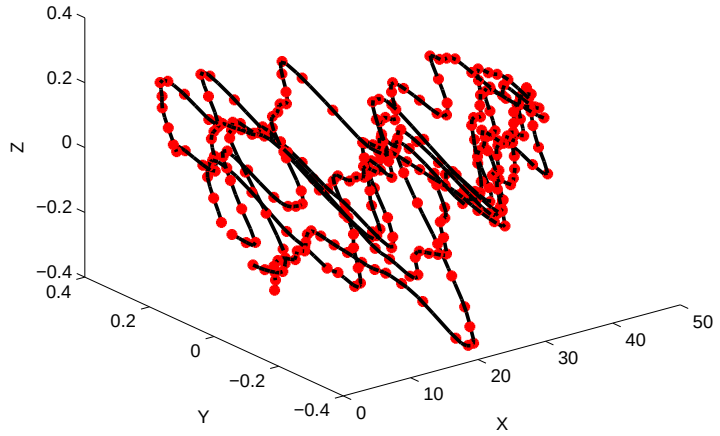
The results suggest almost comparable performance in terms of goodness of fit between the three kernels. However, again, there is a difference in the robustness of the solutions. The Matérn$\frac{3}{2}$ occasionally finds better and worse solutions than both the Matérn$\frac{1}{2}$ and squared exponential kernels, but on average finds the most likely solutions. The results again suggest that the Matérn$\frac{3}{2}$ kernel is the optimal choice for modelling maritime data. It also stands to reason that Matérn$\frac{3}{2}$ kernel is chosen, as this is the kernel associated with the second order differential equations of a standard Newtonian dynamic system. The two directions here are motivated in the same way, hence a Matérn$\frac{3}{2}$ kernel can be applied to either axis, and therefore the data can be described by a multiple output Matérn$\frac{3}{2}$ kernel overall.

6.4 Synthetic Data Illustration

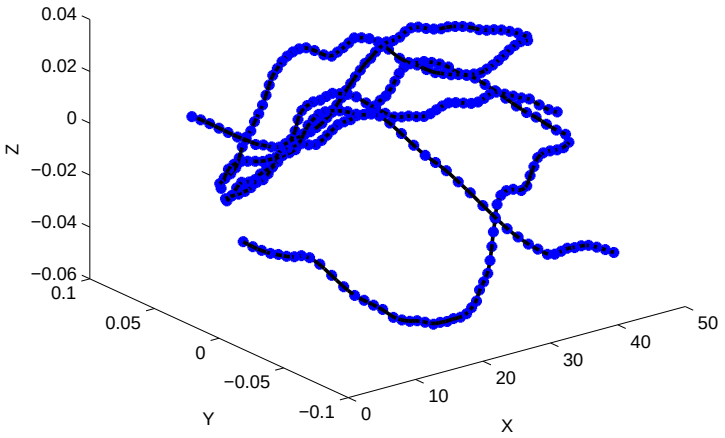
Data was generated from a multiple-output GP using a Matérn $\frac{3}{2}$ kernel with known scaling parameters. For the first 6 generated functions, Figure 6.1a, hyperparameter (Equation 3.22) values of $\epsilon = 5$ and $\lambda = 0.01$ were respectively assigned to the noise and length scale. Additionally the output correlation hyperparameters (Equation 3.25) were assigned values $[\boldsymbol{l}_1]_1 = 0.1$, $[\boldsymbol{l}_2]_1 = 0.1$ and $[\boldsymbol{l}_2]_2 = 0.1$. A further 6 functions were generated from a GP, Figure 6.1b, with hyperparameter values of $\epsilon = 20$, $\lambda = 0.01$, $[\boldsymbol{l}_1]_1 = 0.02$, $[\boldsymbol{l}_2]_1 = 0.02$ and $[\boldsymbol{l}_2]_2 = 0.02$. Inferring hyperparameter values from the generated functions and application

of the Hellinger distance metric between tracks provides for the creation of a similarity matrix, Figure 6.2. This allows a clear split in the data to be identified, in contrast to an examination of the square distance between the kernels alone. The resulting matrix summarising the square distance between each covariance is illustrated in Figure 6.3.

Inversion of the matrix of Hellinger distances between functions satisfies the requirement of a measure of adjacency between nodes within the network. With prior knowledge that two communities exist within the network NMF community detection correctly identifies functions 1 through 6 (generated from the first lot of hyperparameter values) as being from one community, and functions 7 through to 12 as belonging to a distinctly separate community.



(a) Functions generated from a multiple-output GP with hyperparameter values of $\epsilon = 5$, $\lambda = 0.01$, $[\boldsymbol{l}_1]_1 = 0.1$, $[\boldsymbol{l}_2]_1 = 0.1$ and $[\boldsymbol{l}_2]_2 = 0.1$. The mean function using the inferred hyperparameter values is shown plotted through the sample data.



(b) Functions generated from a multiple-output GP with hyperparameter values of $\epsilon = 20$, $\lambda = 0.01$, $[\boldsymbol{l}_1]_1 = 0.02$, $[\boldsymbol{l}_2]_1 = 0.02$ and $[\boldsymbol{l}_2]_2 = 0.02$. The mean function using the inferred hyperparameter values is shown plotted through the sample data.

Figure 6.1

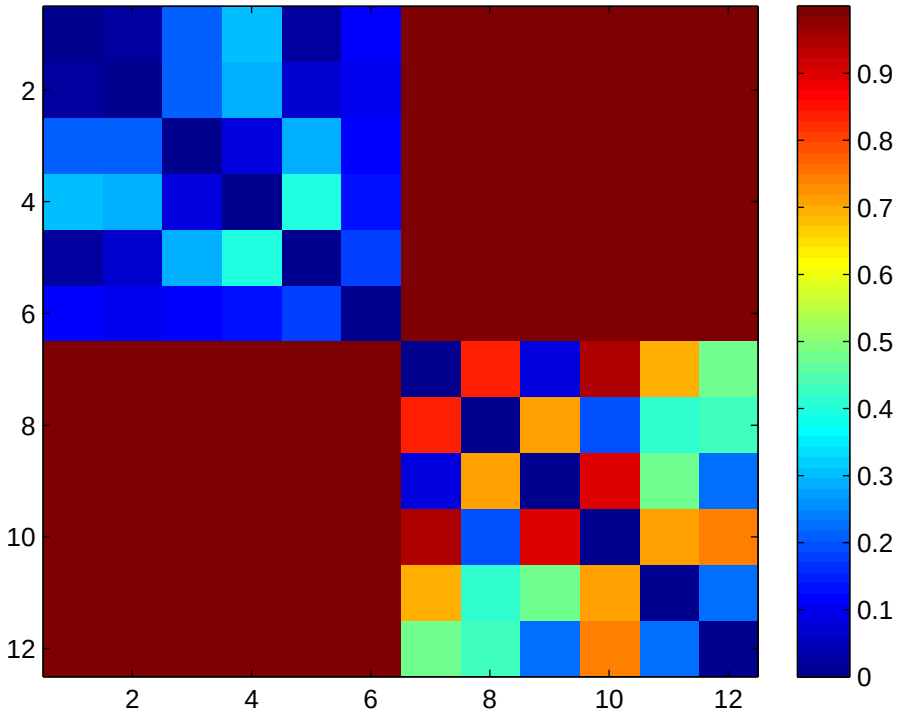


Figure 6.2: Matrix of the Hellinger distance between inferred functions. The axis of the matrix correspond to (in order) functions 1 through 6 (generated from the first set of hyperparameter values) and then functions 7 through 12 (generated from the second set of hyperparameter values). Functions which are very similar to one another have a Hellinger distance close to 0 and further apart close to 1, this is colour illustrated via the colour bar to the right of the Figure.

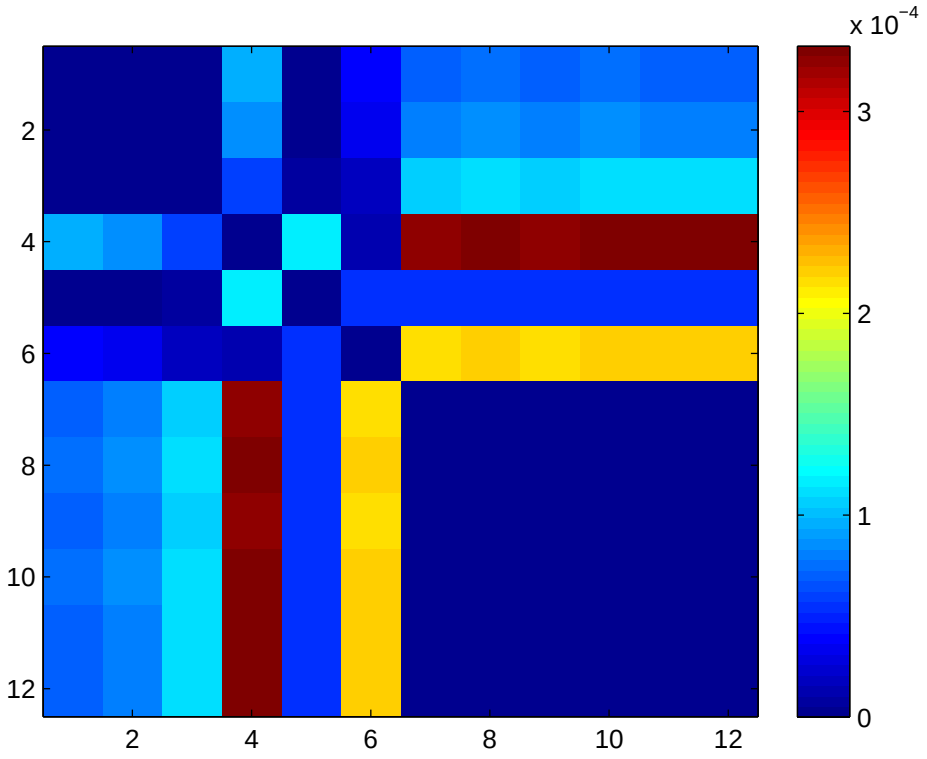


Figure 6.3: Matrix of the square distance between functions. The axis of the matrix correspond to (in order) functions 1 through 6 (generated from the first set of hyperparameter values) and then functions 7 through 12 (generated from the second set of hyperparameter values). Functions which are very dissimilar to one another have distance close to 0, increasing as the functions are deemed further apart by the sum of square distances between kernels. The distance between functions is colour illustrated via the colour bar to the right of the Figure.

6.5 Community Detection in Vessel Track Data

The outlined methodology was also applied to 12 real-world vessel tracks. Each track consists of a set of collected AIS data for the vessel. As such the data collected can be used to evaluate the methodology due to possession of the ground truth. The tracks consisted of four cargo vessel tracks as shown in Figure 6.4, four fishing vessel tracks as shown in Figure 6.5, and four sailing vessel tracks as shown in Figure 6.6. The Hellinger distance between tracks is shown in Figure 6.7. The resulting community structure from application of the GP-NMF technique is shown in Figure 6.8.

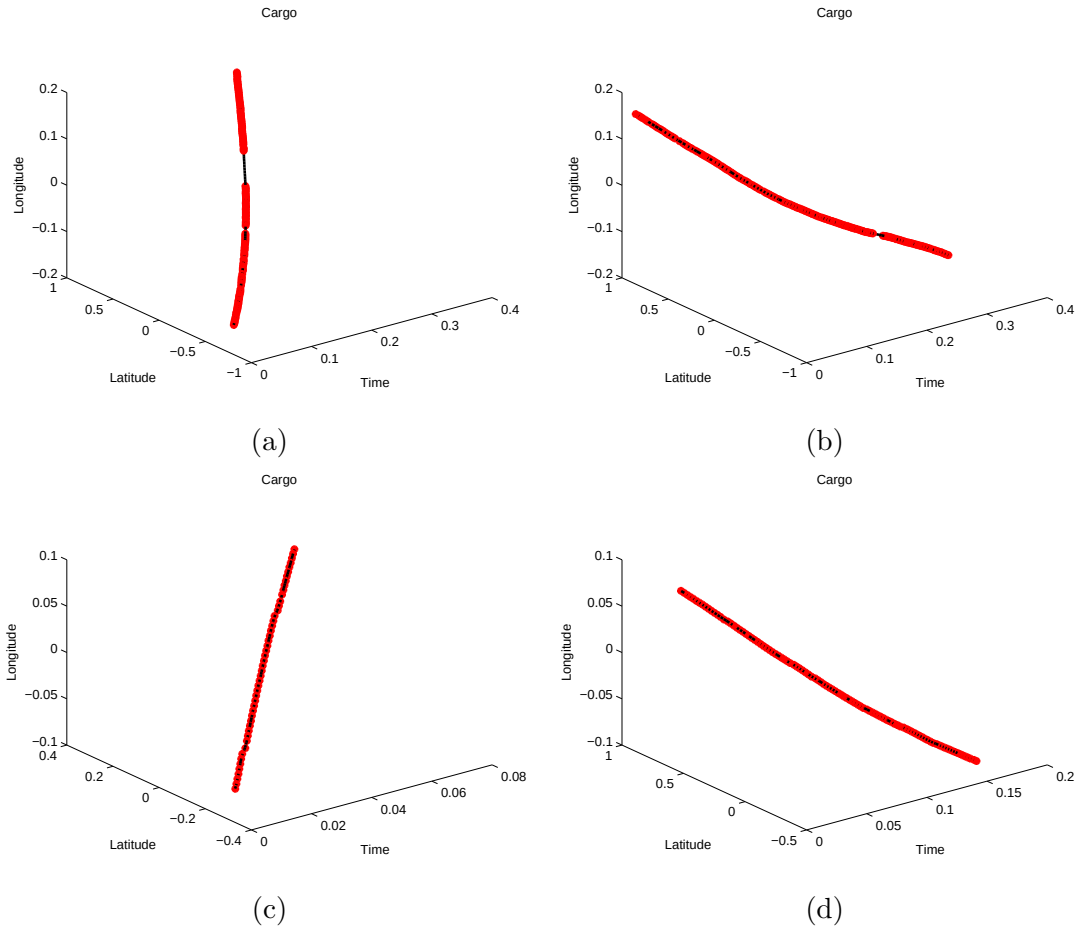


Figure 6.4: Gaussian process multiple output regression through cargo vessel GPS coordinates.

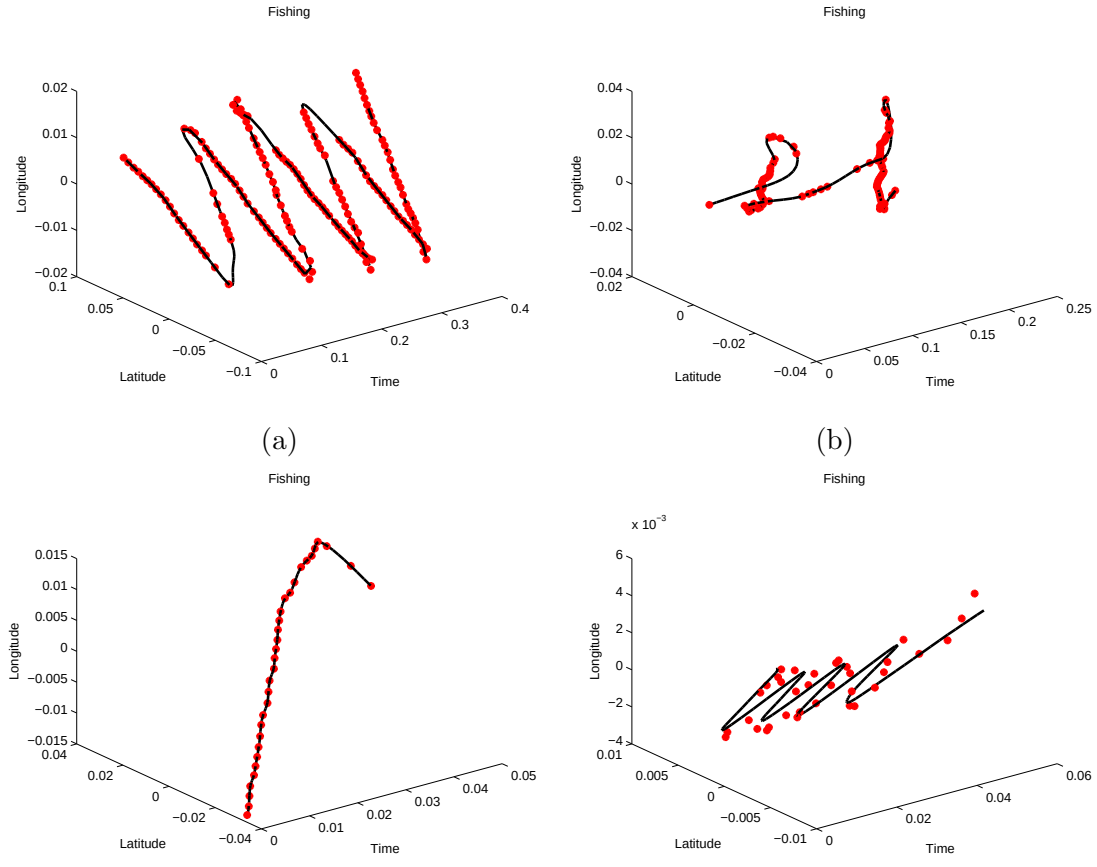


Figure 6.5: Gaussian process multiple output regression through fishing vessel GPS coordinates.

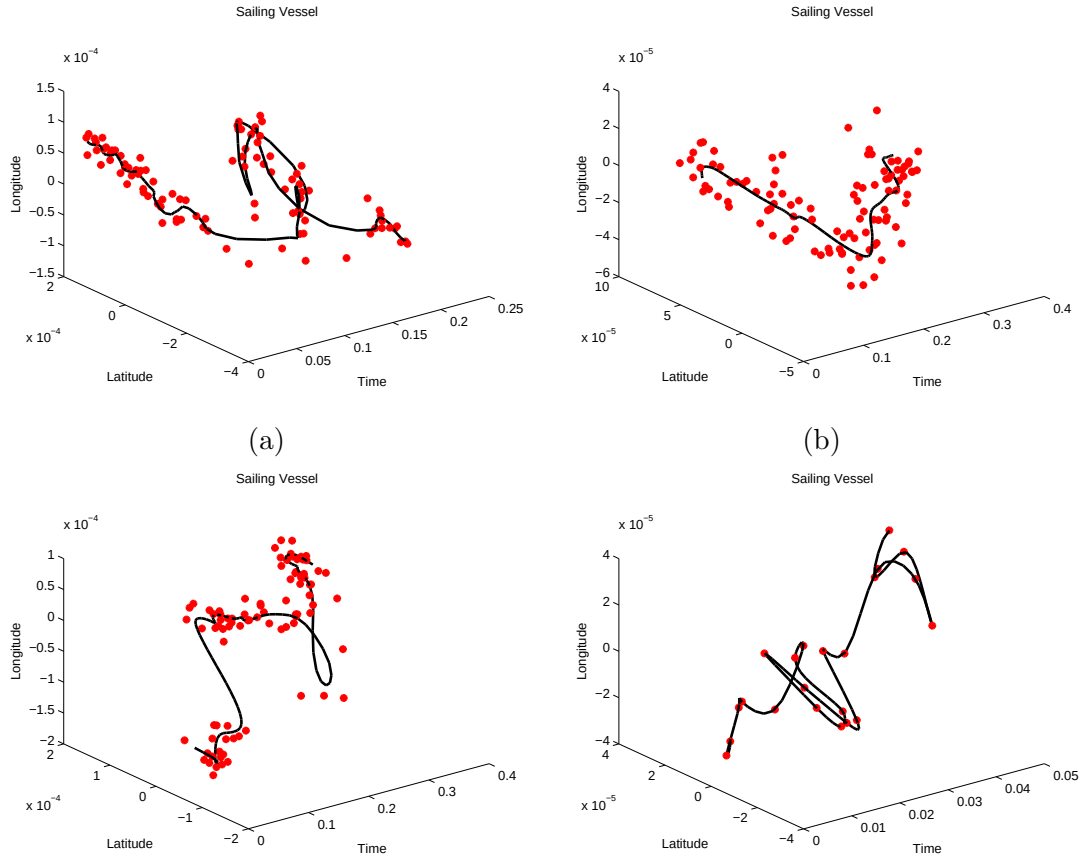


Figure 6.6: Gaussian process multiple output regression through sailing vessel GPS coordinates.

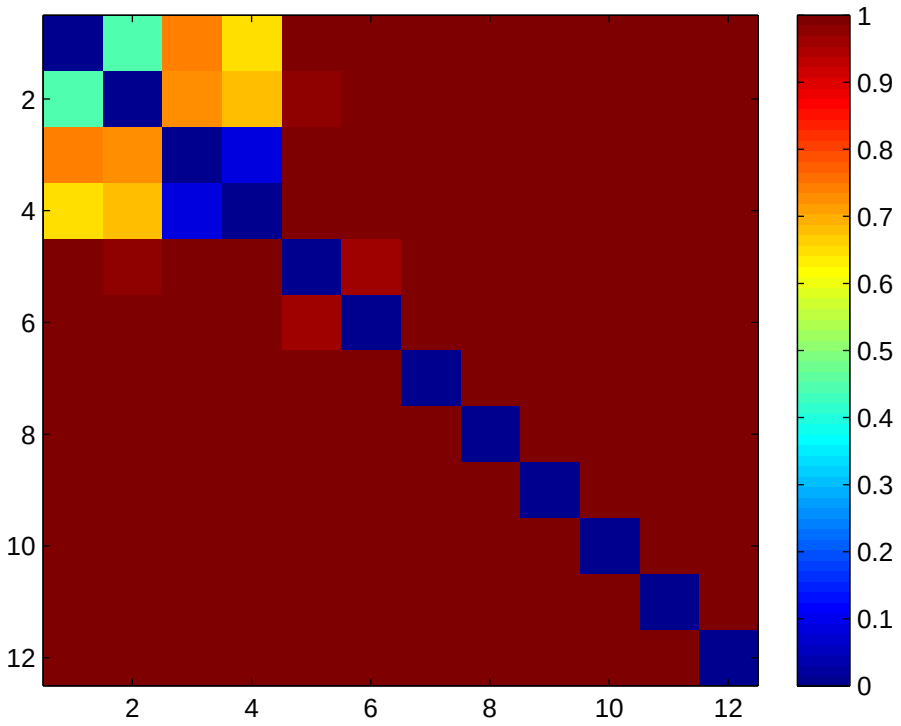


Figure 6.7: Matrix of the Hellinger distance between inferred functions for the vessel data. The axis of the matrix corresponds to (in order) the function inferred for cargo vessel tracks (a) through (d), then the function inferred for fishing vessel tracks (a) through (d). Finally the function inferred for sailing vessel tracks (a) through (d). Functions which are very dissimilar to one another have a Hellinger distance close to 1 and further apart close to zero, this is colour illustrated via the colour bar to the right of the Figure.

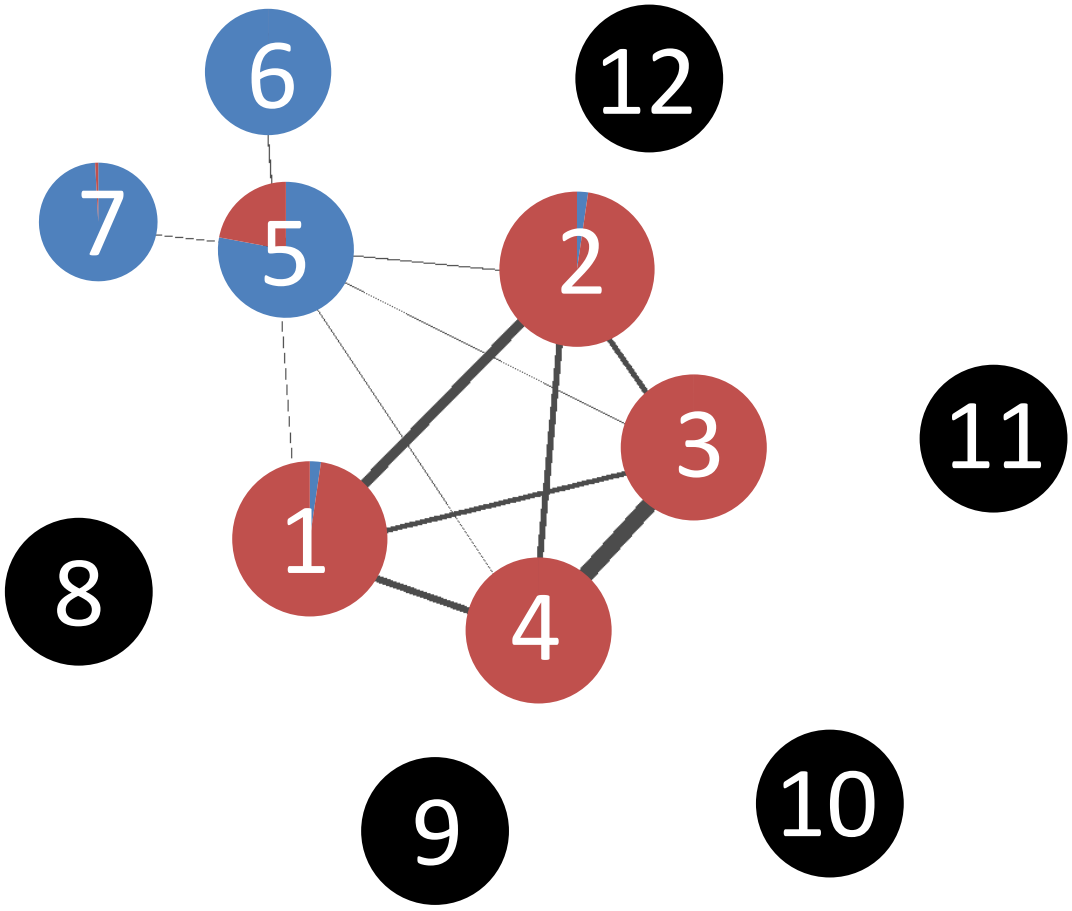


Figure 6.8: Network diagram illustrating the relationship between different vessel types. Each edge connection is weighted (illustrated by the varying edge thickness), the weights being determined by the inverse Hellinger distance between them. Nodes 1 to 12 relate to the different classes of vessel, cargo (1-4) Figure 6.4, fishing (5-8) Figure 6.5 and sailing (9-12) Figure 6.6. Different colours relate to the different communities within the data, and the membership score of each node per community is defined by the pie-chart. The black nodes are unassigned and do not belong to a given community.

Whilst the methodology has clearly identified one distinct community belonging to the cargo vessels the other communities tend to overlap greatly. This is due to the similarity of the inferred functions for the class of cargo vessels, causing the distance between functions to be small. It is postulated that the similarity between functions may be as a result of the restricted area in which the vessels are being forced to operate, i.e. the channel and shipping zones enforcing operating

conditions upon the vessel. This is in addition to the limited track information; AIS signals are only broadcast every four minutes. Even if all subsequent broadcasts are received this might not provide detailed enough information to allow accurate inference of the underlying dynamics. A matter which could only be investigated further if considered in the light of more detailed track information, obtained for instance from RADAR data.

6.6 Summary

The main contribution of the work detailed in this chapter is the creation of a novel distance metric, that when used with Gaussian processes provides a means of expressing the distance between functions. An application of the distance metric to the identification of anomalous vessel tracks (in the tracks entirety) has also been demonstrated. This was achieved by clustering vessels (modelling communities) based on the underlying dynamics mandated by a vessel's class (i.e. fishing vessel, cargo vessel etc.). Modelling of the data was first achieved through application of a multiple-output Gaussian process to a vessel's latitudinal and longitudinal data, ensuring vessel dynamics are suitably captured. The Hellinger distance between the inferred Gaussian process models was then used as a measure between vessel tracks, which can also be interpreted as a distance between functions. Bayesian non-negative matrix factorization was then used as an unsupervised means of identifying community structure, capturing communities of vessels with similar dynamics. It would then be possible to identify anomalies by noting those tracks not assigned to a community. Additionally, subsequent anomalous vessel tracks could be detected by noting discrepancies between the vessel class broadcast by the vessel (as mandated in Automatic Identification System data) and the labelled community to which the track is assigned. The results from applying the technique to a small data set of 12 tracks clearly identify one distinct community and a further two overlapping communities. The overlapping nature of the results indicate that further analysis is required in order to fully evaluate the applicability of this technique to maritime abnormality detection. Whilst we would intuitively expect, given the variability of vessel types belonging to a vessel class, that there is some overlap between communities it

could only be proven through empirical investigation whether there is sufficient similarity between vessel classes to form distinct community structures.

Chapter 7

Conclusion and Future Work

This thesis began by identifying the current state of the art in issues surrounding marine abnormality detection. During the course of the review several failings of existing methods were identified, these have included noting that current domain methods have lacked a principled approach to testing for anomalies, failing to account for outlying extreme observations. Furthermore the flexible Gaussian process framework which allows the relationship between inputs to be expressed in a non-parametric form has seen limited application within this domain.

Modelling accuracy was first investigated through investigation of a suitable kernel function to accurately model vessel dynamics. Although, the choice of Gaussian process kernel function becomes less critical with increasing amounts of data, for smaller sample sizes, the kernel function is critical. A novel kernel was derived based on the Kalman filter near constant velocity model and was compared to the standard squared exponential, Matérn$\frac{3}{2}$ and Matérn$\frac{1}{2}$ kernels. The near constant velocity kernel will no doubt offer improved modelling performance under certain scenarios, but investigation proved the Matérn$\frac{3}{2}$ kernel optimal for modelling the feature space representation of vessel tracks.

The power and flexibility of the Gaussian process framework does however come at a computational cost, namely the inversion of the covariance matrix. By demonstrating how the sequential update of the Gaussian Process covariance matrix can be achieved the need for inverting massive matrices substantially reduces the computational burdens for which Gaussian processes are well known. Thus this work has shown how Gaussian processes can be applied to the marine

domain in a manner which is computationally viable.

Extreme value theory has also proven to be an extremely successful framework for anomaly detection. Unlike novelty detection based directly on the sample distribution, extreme value distributions capture the belief that extreme events should become more extreme if large numbers of measurements are expected and vice versa. Such detection, however, has to be dynamic, context sensitive and timely if it is to be useful for marine tracking. Extreme value distributions alone are not readily adapted to perform this task.

In this thesis extreme value distributions were endowed directly with dynamic properties, allowing the creation of a dynamic bound, this was achieved through adapting the underlying probability density function based on the data sampling rate. Through a novel combination with Gaussian processes it has been shown how the true underlying form of predictive distributions can be calculated in a principled manner, leading to improved abnormality detection within vessel tracks.

Empirical experiments on vessel data suggest that the Gaussian process-extreme value theory technique is capable of detecting anomalies that resemble mooring or drifting, and unexpected departures from regular movements. The sample size prediction plays the important role of adapting the observation process in time. As the effective sample size reduces, the extreme value distribution approaches the regular Gaussian distribution, as Equation 5.1 suggests. However, with increasing density of observations the extreme value distribution diverges and the extreme value theory bound increases.

The representative choice of distance as the dependent variable feature for anomaly detection is open to discussion. While it provides a single dimension and, thus, fast estimation it does fail to capture some aspects of ship tracks. To capture such features the GPS coordinates could be simultaneously modelled with a bivariate Gaussian Process and extrema modelling.

However it can be noted that the detection of outlying points using these methods is dependent on the feature space. In order to move away from this dependence techniques from the field of divergence measurement were applied. Specifically the distance between distributions taking a Gaussian form were determined using the Hellinger distance. Under the Gaussian process framework functions are de-

scribed by a multivariate Gaussian, application in this manner therefore becomes a distance between functions. In this manner communities of vessels can be determined and anomalies identified. Anomalies then relate to outlying functions and not to the feature space used to transform the data.
To provide a better representation of the vessel motion patterns the tracks were modelled using a bivariate Gaussian process. Thus the dynamics which relate to a vessels motion patterns were more accurately captured through considering the relationship between latitude and longitude and not casting these motions into a different feature space. The feature space would only serve to mask the dynamics exhibited by different classes of vessel.
The detection of communities was undertaken using unsupervised techniques based on non-negative matrix factorisation. Thus it was demonstrated how vessel tracks could be classified based on the underlying dynamics of the vessel track, and used as a means of vessel class identification. Abnormalities were identified based on the incorrect classification of vessels class and the community to which it was assigned.
Empirical experiments suggest that whilst the method is capable of discerning between classes of vessel and identifying community structure according to their vessel type, this is not always accurate. This is due to the similarity in the inferred functions and correspondingly the distance between inferred functions tending to be very small. This may have been as a result of vessels being forced to operate under a given set of operating conditions which mandate their movement, such as in a shipping lane, forcing their functions to take a restrictive form. Vessels deviating from the constrained conditions may have a distinctly different functional form which may be identified as belonging to a different community structure. This may be in addition to the limited time stamped spatial track information. These are all areas which need further investigation.

7.1 Future Work

This work has laid the foundations for many avenues of future research and work. The track abnormality detection using Gaussian processes and extreme value theory can be extended to detect anomalies in multi-dimensional data, possibly

detecting marine abnormalities based on the vessels latest latitude and longitude. It could furthermore be applied to shipping lanes and vessel types. This would allow anomaly detection not just on the basis of individual points but entire tracks, and so offers the possibility of preventing accidents such as that of the MS Costa Concordia early in 2012. Kernel-regression based prediction of the sample size can be readily extended using Poisson processes.

The work on community detection based on the underlying vessel dynamics is also open to many further avenues of expansion and investigation. A more appropriate application of the work may be in identifying deviations from typical shipping routes. Also fusion with additional data sources such as from HF RADAR may allow a more detailed model of the vessel track to be constructed, and in so doing improve the community detection process. Communities may also be more accurately detected through consideration of a supervised learning process.

Appendix A - Partitioning the multivariate Gaussian distribution

The multivariate Gaussian distribution offers many properties that can be exploited, commonly being parameterised for a n dimensional random vector $\boldsymbol{x} \in \mathbb{R}$ as,

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^{\top} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right). \tag{1}$$

The random vector $\boldsymbol{x}$, mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ can be partitioned according to,

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x_a} \\ \boldsymbol{x_b} \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu_a} \\ \boldsymbol{\mu_b} \end{pmatrix} \text{ and } \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma_{aa}} & \boldsymbol{\Sigma_{ab}} \\ \boldsymbol{\Sigma_{ba}} & \boldsymbol{\Sigma_{bb}} \end{pmatrix}. \tag{2}$$

This concept of partitioning a matrix introduces a useful concept, in that we can consider the matrix $\boldsymbol{x}$ as consisting of distinct sets of data, $\boldsymbol{x}_a$ could be data relating to an observed data set and $\boldsymbol{x}_b$ could be data relating to a prediction set. It is sometimes useful to express $\boldsymbol{\Sigma}$ in terms of the precision, $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$, in which case the partitioning can be expressed as,

$$\begin{pmatrix} \boldsymbol{\Sigma_{aa}} & \boldsymbol{\Sigma_{ab}} \\ \boldsymbol{\Sigma_{ba}} & \boldsymbol{\Sigma_{bb}} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda_{aa}} & \boldsymbol{\Lambda_{ab}} \\ \boldsymbol{\Lambda_{ba}} & \boldsymbol{\Lambda_{bb}} \end{pmatrix}. \tag{3}$$

It should however be noted that,

$$\begin{pmatrix} \boldsymbol{\Lambda_{aa}} & \boldsymbol{\Lambda_{ab}} \\ \boldsymbol{\Lambda_{ba}} & \boldsymbol{\Lambda_{bb}} \end{pmatrix}^{-1} \neq \begin{pmatrix} \boldsymbol{\Lambda_{aa}}^{-1} & \boldsymbol{\Lambda_{ab}}^{-1} \\ \boldsymbol{\Lambda_{ba}}^{-1} & \boldsymbol{\Lambda_{bb}}^{-1} \end{pmatrix}. \tag{4}$$

To convert between the inverse of the individually partitioned terms and the inverse of the entire matrix,

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{M}^{-1} & -\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}^{-1} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \end{pmatrix}, \tag{5}$$

where $\boldsymbol{M} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$. This formula can be thought of as the multivariate generalisation of the explicit inverse for a 2×2 matrix,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}. \tag{6}$$

Equation 3 and Equation 5 allow the relation between inverse covariance and precision to be expressed as,

$$\begin{pmatrix} \boldsymbol{\Sigma_{aa}} & \boldsymbol{\Sigma_{ab}} \\ \boldsymbol{\Sigma_{ba}} & \boldsymbol{\Sigma_{bb}} \end{pmatrix} = \begin{pmatrix} \left(\boldsymbol{\Lambda_{aa}} - \boldsymbol{\Lambda_{ab}}\boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\right)^{-1} & -\left(\boldsymbol{\Lambda_{aa}} - \boldsymbol{\Lambda_{ab}}\boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\right)^{-1}\boldsymbol{\Lambda_{ab}}\boldsymbol{\Lambda_{bb}}^{-1} \\ -\boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{\Lambda_{aa}} - \boldsymbol{\Lambda_{ab}}\boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\right)^{-1} & \left(\boldsymbol{\Lambda_{bb}} - \boldsymbol{\Lambda_{ba}}\boldsymbol{\Lambda_{aa}}^{-1}\boldsymbol{\Lambda_{ab}}\right)^{-1} \end{pmatrix}. \tag{7}$$

Appendix B - Conditional and marginal distributions of the multivariate Gaussian

If the random vector $\boldsymbol{x} \in \mathbb{R}$ is Gaussian distributed according to Equation 1 of Appendix A, and partitioned according to Equation 2 of Appendix A, then the joint distribution $p(\boldsymbol{x_a}, \boldsymbol{x_b})$ can be expressed as a sum of Gaussians,

$$\begin{aligned} p(\boldsymbol{x_a}, \boldsymbol{x_b}) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) = \\ \frac{1}{Z}\exp\Big(-\frac{1}{2}(\boldsymbol{x_a}-\boldsymbol{\mu_a})^\top \boldsymbol{\Lambda_{aa}}(\boldsymbol{x_a}-\boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_a}-\boldsymbol{\mu_a})^\top \boldsymbol{\Lambda_{ab}}(\boldsymbol{x_b}-\boldsymbol{\mu_b}) \\ -\frac{1}{2}(\boldsymbol{x_b}-\boldsymbol{\mu_b})^\top \boldsymbol{\Lambda_{ba}}(\boldsymbol{x_a}-\boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_b}-\boldsymbol{\mu_b})^\top \boldsymbol{\Lambda_{bb}}(\boldsymbol{x_b}-\boldsymbol{\mu_b})\Big), \end{aligned} \tag{8}$$

where $Z = (2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Sigma}|}$, which is a constant that does not depend on $\boldsymbol{x_a}$ or $\boldsymbol{x_b}$. This term ensures the quadratic form normalises. To obtain the marginal distribution $p(\boldsymbol{x_a})$, the $\boldsymbol{x_b}$ terms need to be integrated out of the summation,

$$\begin{aligned} p(\boldsymbol{x_a}) = \frac{1}{Z}\int \exp\Big(-\Big[\frac{1}{2}(\boldsymbol{x}_a-\boldsymbol{\mu}_a)^\top \boldsymbol{\Lambda_{aa}}(\boldsymbol{x}_a-\boldsymbol{\mu}_a) + \frac{1}{2}(\boldsymbol{x}_a-\boldsymbol{\mu}_a)^\top \boldsymbol{\Lambda_{ab}}(\boldsymbol{x}_b-\boldsymbol{\mu}_b) \\ +\frac{1}{2}(\boldsymbol{x}_b-\boldsymbol{\mu}_b)^\top \boldsymbol{\Lambda_{ba}}(\boldsymbol{x}_a-\boldsymbol{\mu}_a) + \frac{1}{2}(\boldsymbol{x}_b-\boldsymbol{\mu}_b)^\top \boldsymbol{\Lambda_{bb}}(\boldsymbol{x}_b-\boldsymbol{\mu}_b)\Big]\Big) d\boldsymbol{x_b} \end{aligned} \tag{9}$$

To evaluate the integral it is beneficial for the expression within the exponent to be expressed in an alternate form. The technique known as completing the square provides a means of equating a quadratic polynomial, which is of the form $\boldsymbol{z}^\top \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}^\top \boldsymbol{z} + \boldsymbol{c}$ and $\boldsymbol{A}$ a symmetric non-singular matrix, to a quadratic form,

$$\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}^\top \boldsymbol{z} + \boldsymbol{c} = \frac{1}{2}\left(\boldsymbol{z} + \boldsymbol{A}^{-1}\boldsymbol{b}\right)^\top \boldsymbol{A}\left(\boldsymbol{z} + \boldsymbol{A}^{-1}\boldsymbol{b}\right) + \boldsymbol{c} - \frac{1}{2}\boldsymbol{b}^\top \boldsymbol{A}^{-1}\boldsymbol{b}. \tag{10}$$

The technique is a generalisation of the method used in single variable algebra,

$$\frac{1}{2}az^2 + bz + c = \frac{1}{2}a\left(z + \frac{b}{a}\right)^2 + c - \frac{b^2}{2a}. \tag{11}$$

To apply the completion of squares to Equation 9 let,

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{x_b} - \boldsymbol{\mu_b}, & \boldsymbol{A} &= \boldsymbol{\Lambda_{bb}}, \\ \boldsymbol{b} &= \boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right), & \boldsymbol{c} &= \frac{1}{2}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right)^\top \boldsymbol{\Lambda_{aa}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right). \end{aligned} \tag{12}$$

The integral can now be rewritten,

$$\begin{aligned} p\left(\boldsymbol{x_a}\right) = \frac{1}{Z}\int \exp\Bigg(- \Bigg[& \frac{1}{2}\left(\boldsymbol{x_b} - \boldsymbol{\mu_b} + \boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right)\right)^\top \\ & \boldsymbol{\Lambda_{bb}}\left(\boldsymbol{x_b} - \boldsymbol{\mu_b} + \boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right)\right) \\ & + \frac{1}{2}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right)^\top \boldsymbol{\Lambda_{aa}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right) \\ & - \frac{1}{2}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right)^\top \boldsymbol{\Lambda_{ab}}\boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right) \Bigg] \Bigg) d\boldsymbol{x_b}. \end{aligned} \tag{13}$$

Constants can also be factored out and included within Z,

$$\begin{aligned} p\left(\boldsymbol{x_a}\right) = & \frac{1}{Z}\int \exp\Bigg(- \frac{1}{2}\left(\boldsymbol{x_b} - \boldsymbol{\mu_b} + \boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right)\right)^\top \\ & \boldsymbol{\Lambda_{bb}}\left(\boldsymbol{x_b} - \boldsymbol{\mu_b} + \boldsymbol{\Lambda_{bb}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a} - \boldsymbol{\mu_a}\right)\right) \Bigg) d\boldsymbol{x_b}. \end{aligned} \tag{14}$$

By noting that the density function must normalise in order to ensure consistency

the integral can be equated to a given analytic form,

$$\begin{aligned}\frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Sigma}|}}\int \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right)d\boldsymbol{x}=1,\\ \int \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right)d\boldsymbol{x}=(2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Sigma}|}.\end{aligned} \tag{15}$$

The integral in the marginal distribution can then be solved by equating its form to Equation 15,

$$\begin{aligned}(2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Sigma}|}=\int \exp\Big(-\frac{1}{2}\left(\boldsymbol{x_b}-\boldsymbol{\mu_b}+{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\right)^{\top}\\ \boldsymbol{\Lambda_{bb}}\left(\boldsymbol{x_b}-\boldsymbol{\mu_b}+{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\right)\Big)d\boldsymbol{x_b}.\end{aligned} \tag{16}$$

The marginal probability can therefore be expressed as,

$$p\left(\boldsymbol{x_a}\right)=\frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Lambda_{bb}}|}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)^{\top}\left(\boldsymbol{\Lambda_{aa}}-\boldsymbol{\Lambda_{ab}}{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\right)\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\right). \tag{17}$$

Equating the terms in Equation 17 to Equation 7 it is possible to state the marginal probability in terms of its mean and covariance,

$$\begin{aligned}p\left(\boldsymbol{x_a}\right)&=\mathcal{N}\left(\boldsymbol{\mu_a},\boldsymbol{\Sigma_{aa}}\right),\\ p\left(\boldsymbol{x_b}\right)&=\mathcal{N}\left(\boldsymbol{\mu_b},\boldsymbol{\Sigma_{bb}}\right).\end{aligned} \tag{18}$$

The same approach can also be taken to calculate the conditional distributions $p\left(\boldsymbol{x_a}|\boldsymbol{x_b}\right)$ and $p\left(\boldsymbol{x_b}|\boldsymbol{x_a}\right)$. From the product rule of probability, it can be noted that the conditional probability can be found by normalising with respect to the

marginal distribution,

$$
\begin{aligned}
p\left(\boldsymbol{x_a}|\boldsymbol{x_b}\right) &= \frac{p\left(\boldsymbol{x_a},\boldsymbol{x_b};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)}{\int p\left(\boldsymbol{x_a},\boldsymbol{x_b};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)d\boldsymbol{x_a}}, \\
p\left(\boldsymbol{x_b}|\boldsymbol{x_a}\right) &= \frac{p\left(\boldsymbol{x_a},\boldsymbol{x_b};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)}{\int p\left(\boldsymbol{x_a},\boldsymbol{x_b};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)d\boldsymbol{x_b}}.
\end{aligned}
\tag{19}
$$

Then it follows that,

$$
\begin{aligned}
p\left(\boldsymbol{x_b}|\boldsymbol{x_a}\right) = \frac{1}{Z}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) = \\
\frac{1}{Z}\exp\Bigg(-\frac{1}{2}\left(\boldsymbol{x}_a-\boldsymbol{\mu}_a\right)^\top\boldsymbol{\Lambda}_{aa}\left(\boldsymbol{x}_a-\boldsymbol{\mu}_a\right)-\frac{1}{2}\left(\boldsymbol{x}_a-\boldsymbol{\mu}_a\right)^\top\boldsymbol{\Lambda}_{ab}\left(\boldsymbol{x}_b-\boldsymbol{\mu}_b\right) \\
-\frac{1}{2}\left(\boldsymbol{x}_b-\boldsymbol{\mu}_b\right)^\top\boldsymbol{\Lambda}_{ba}\left(\boldsymbol{x}_a-\boldsymbol{\mu}_a\right)-\frac{1}{2}\left(\boldsymbol{x}_b-\boldsymbol{\mu}_b\right)^\top\boldsymbol{\Lambda}_{bb}\left(\boldsymbol{x}_b-\boldsymbol{\mu}_b\right)\Bigg),
\end{aligned}
\tag{20}
$$

where Z and has absorbed $p\left(\boldsymbol{x_a}\right)$, terms which do not depend on $\boldsymbol{x_b}$. This can again be simplified using the terms in Equation 12 and completing the square,

$$
\begin{aligned}
p\left(\boldsymbol{x_b}|\boldsymbol{x_a}\right) = \frac{1}{Z}\exp\Bigg(-\Bigg[\frac{1}{2}\left(\boldsymbol{x_b}-\boldsymbol{\mu_b}+{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\right)^\top \\
\boldsymbol{\Lambda_{bb}}\left(\boldsymbol{x_b}-\boldsymbol{\mu_b}+{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\right) \\
+\frac{1}{2}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)^\top\boldsymbol{\Lambda_{aa}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right) \\
-\frac{1}{2}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)^\top\boldsymbol{\Lambda_{ab}}{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\Bigg]\Bigg).
\end{aligned}
\tag{21}
$$

Further terms which do not depend on $\boldsymbol{x_b}$ can be additionally captured within the Z term,

$$
\begin{aligned}
p\left(\boldsymbol{x_b}|\boldsymbol{x_a}\right) = \frac{1}{Z}\exp\Bigg(-\frac{1}{2}\left(\boldsymbol{x_b}-\boldsymbol{\mu_b}+{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\right)^\top \\
\boldsymbol{\Lambda_{bb}}\left(\boldsymbol{x_b}-\boldsymbol{\mu_b}+{\boldsymbol{\Lambda_{bb}}}^{-1}\boldsymbol{\Lambda_{ba}}\left(\boldsymbol{x_a}-\boldsymbol{\mu_a}\right)\right)\Bigg).
\end{aligned}
\tag{22}
$$

By again equating terms to Equation 7 it is possible to state,

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{b}|\boldsymbol{a}} &= \boldsymbol{\mu}_{\boldsymbol{b}} - \boldsymbol{\Lambda}_{\boldsymbol{bb}}{}^{-1}\boldsymbol{\Lambda}_{\boldsymbol{ba}}\left(\boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{\mu}_{\boldsymbol{a}}\right) = \boldsymbol{\mu}_{\boldsymbol{b}} + \boldsymbol{\Sigma}_{\boldsymbol{ba}}\boldsymbol{\Sigma}_{\boldsymbol{aa}}{}^{-1}\left(\boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{\mu}_{\boldsymbol{a}}\right), \\ \boldsymbol{\Sigma}_{\boldsymbol{b}|\boldsymbol{a}} &= \boldsymbol{\Lambda}_{\boldsymbol{bb}}{}^{-1} = \boldsymbol{\Sigma}_{\boldsymbol{bb}} - \boldsymbol{\Sigma}_{\boldsymbol{ba}}\boldsymbol{\Sigma}_{\boldsymbol{aa}}{}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{ab}}.\end{aligned} \tag{23}$$

It is therefore possible to express the conditional distributions in terms of a mean and covariance,

$$\begin{aligned}p\left(\boldsymbol{x}_{\boldsymbol{a}}|\boldsymbol{x}_{\boldsymbol{b}}\right) &\sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{a}} + \boldsymbol{\Sigma}_{\boldsymbol{ab}}\boldsymbol{\Sigma}_{\boldsymbol{bb}}{}^{-1}\left(\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{\mu}_{\boldsymbol{b}}\right), \boldsymbol{\Sigma}_{\boldsymbol{aa}} - \boldsymbol{\Sigma}_{\boldsymbol{ab}}\boldsymbol{\Sigma}_{\boldsymbol{bb}}{}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{ba}}\right), \\ p\left(\boldsymbol{x}_{\boldsymbol{b}}|\boldsymbol{x}_{\boldsymbol{a}}\right) &\sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{b}} + \boldsymbol{\Sigma}_{\boldsymbol{ba}}\boldsymbol{\Sigma}_{\boldsymbol{aa}}{}^{-1}\left(\boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{\mu}_{\boldsymbol{a}}\right), \boldsymbol{\Sigma}_{\boldsymbol{bb}} - \boldsymbol{\Sigma}_{\boldsymbol{ba}}\boldsymbol{\Sigma}_{\boldsymbol{aa}}{}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{ab}}\right).\end{aligned} \tag{24}$$

Noting from Equation 23 that the mean of a conditional distribution can be expressed as a linear function and the covariance is independent of $\boldsymbol{x}$, then the conditional distribution $p\left(\boldsymbol{x}_{\boldsymbol{b}}|\boldsymbol{x}_{\boldsymbol{a}}\right)$ can also be expressed as,

$$p\left(\boldsymbol{x}_{\boldsymbol{b}}|\boldsymbol{x}_{\boldsymbol{a}}\right) = \mathcal{N}\left(\boldsymbol{x}_{\boldsymbol{b}}; \boldsymbol{M}\boldsymbol{x}_{\boldsymbol{a}} + \boldsymbol{b}, \boldsymbol{\Sigma}_{\boldsymbol{b}|\boldsymbol{a}}\right). \tag{25}$$

Bayes theorem (Equation 3.2) can now be applied to derive the posterior distribution $p\left(\boldsymbol{x}_{\boldsymbol{a}}|\boldsymbol{x}_{\boldsymbol{b}}\right)$. The desired operation is dependent on obtaining the marginal distribution $p\left(\boldsymbol{x}_{\boldsymbol{b}}\right)$. This can be obtained by using the product rule to express the joint distribution (x_a, x_b) as product of $p\left(\boldsymbol{x}_{\boldsymbol{b}}|\boldsymbol{x}_{\boldsymbol{a}}\right)$ given in Equation 25, and prior distribution $p\left(\boldsymbol{x}_{\boldsymbol{a}}\right)$, of which the assumed form is known. To marginalise unwanted parameters the joint distribution can be partitioned, from which the form of the marginal distribution can be obtained. The joint distribution is therefore expressed as,

$$\begin{aligned}p\left(\boldsymbol{x}_{\boldsymbol{b}}|\boldsymbol{x}_{\boldsymbol{a}}\right)p\left(\boldsymbol{x}_{\boldsymbol{a}}\right) = \frac{1}{Z}\exp\Big(&-\frac{1}{2}\left(\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{M}\boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{b}\right)^{\top}\boldsymbol{\Sigma}_{\boldsymbol{b}|\boldsymbol{a}}{}^{-1}\left(\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{M}\boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{b}\right) \\ &-\frac{1}{2}\left(\boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{\mu}_{\boldsymbol{a}}\right)^{\top}\boldsymbol{\Sigma}_{\boldsymbol{a}}{}^{-1}\left(\boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{\mu}_{\boldsymbol{a}}\right)\Big).\end{aligned} \tag{26}$$

A reduced form can be achieved through introduction of the following variables,

$$\boldsymbol{e} = \boldsymbol{x_a} - \boldsymbol{\mu_a}, \qquad \boldsymbol{f} = \boldsymbol{x_b} - \boldsymbol{M}\boldsymbol{\mu_b} - \boldsymbol{b}. \tag{27}$$

These variables allow the joint distribution to be expressed in quadratic form,

$$= \frac{1}{Z}\exp\left(-\frac{1}{2}\left[(\boldsymbol{f} - \boldsymbol{M}\boldsymbol{e})^\top \boldsymbol{\Sigma_{b|a}}^{-1}(\boldsymbol{f} - \boldsymbol{M}\boldsymbol{e}) + \boldsymbol{e}^\top \boldsymbol{\Sigma_a}^{-1}\boldsymbol{e}\right]\right). \tag{28}$$

Partitioning of the quadratic form, as in Equation 8, allows the result to be expressed as,

$$\begin{aligned} = \frac{1}{Z}\exp\Bigg(-\frac{1}{2}\Bigg[&\boldsymbol{e}^\top\left(\boldsymbol{M}^\top \boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{M} + \boldsymbol{\Sigma_a}^{-1}\right)\boldsymbol{e} \\ &- \boldsymbol{e}^\top \boldsymbol{M}^\top \boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{f} - \boldsymbol{f}\boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{M}\boldsymbol{e} + \boldsymbol{f}^\top \boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{f}\Bigg]\Bigg). \end{aligned} \tag{29}$$

The quadratic form of Equation 29 can alternatively be expressed in its partitioned form,

$$= \frac{1}{Z}\exp\left(-\frac{1}{2}\begin{pmatrix}\boldsymbol{e}\\ \boldsymbol{f}\end{pmatrix}^\top \begin{pmatrix}\boldsymbol{M}^\top \boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{M} + \boldsymbol{\Sigma_a}^{-1} & -\boldsymbol{M}^\top \boldsymbol{\Sigma_{b|a}}^{-1} \\ -\boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{M} & \boldsymbol{\Sigma_{b|a}}^{-1}\end{pmatrix}^{-1}\begin{pmatrix}\boldsymbol{e}\\ \boldsymbol{f}\end{pmatrix}\right). \tag{30}$$

Using Equation 5, the terms used to define the covariance can also be expressed as,

$$\begin{pmatrix}\boldsymbol{M}^\top \boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{M} + \boldsymbol{\Sigma_a}^{-1} & -\boldsymbol{M}^\top \boldsymbol{\Sigma_{b|a}}^{-1} \\ -\boldsymbol{\Sigma_{b|a}}^{-1}\boldsymbol{M} & \boldsymbol{\Sigma_{b|a}}^{-1}\end{pmatrix}^{-1} = \begin{pmatrix}\boldsymbol{\Sigma_a} & \boldsymbol{\Sigma_a}\boldsymbol{M}^\top \\ \boldsymbol{M}\boldsymbol{\Sigma_a} & \boldsymbol{\Sigma_{b|a}} + \boldsymbol{M}\boldsymbol{\Sigma_a}\boldsymbol{M}^\top\end{pmatrix}. \tag{31}$$

The marginal density $p(\boldsymbol{x_b})$ is then given as,

$$p(\boldsymbol{x_b}) = \mathcal{N}\left(\boldsymbol{x_b}; \boldsymbol{M}\boldsymbol{\mu_a} + \boldsymbol{b}, \boldsymbol{\Sigma_{b|a}} + \boldsymbol{M}\boldsymbol{\Sigma_a}\boldsymbol{M}^\top\right). \tag{32}$$

The conditional density $p(\boldsymbol{x_a}|\boldsymbol{x_b})$ can then be obtained by following the previous

steps used to derive the conditional distribution substituting in Equation 32 for the marginal form. The resulting posterior distribution is expressed as,

$$
\begin{aligned}
& p\left(\boldsymbol{x_a}|\boldsymbol{x_b}\right) \\
& = \mathcal{N}\left(\boldsymbol{x_a}; \boldsymbol{\Sigma_{a|b}}\left(\boldsymbol{M}^\top {\boldsymbol{\Sigma_{b|a}}}^{-1}\left(\boldsymbol{x_b}-\boldsymbol{b}\right)+{\boldsymbol{\Sigma_a}}^{-1}\boldsymbol{\mu_a}\right), \left({\boldsymbol{\Sigma_a}}^{-1}+\boldsymbol{M}^\top\boldsymbol{\Sigma_{b|a}}\boldsymbol{M}\right)^{-1}\right) \\
& = \mathcal{N}\left(\boldsymbol{x_a}; \boldsymbol{\mu_a}+\boldsymbol{\Sigma_a}\boldsymbol{M}^\top {\boldsymbol{\Sigma_b}}^{-1}\left(\boldsymbol{x_b}-\boldsymbol{b}-\boldsymbol{M}\boldsymbol{\mu_a}\right), \boldsymbol{\Sigma_a}-\boldsymbol{\Sigma_a}\boldsymbol{M}^\top {\boldsymbol{\Sigma_b}}^{-1}\boldsymbol{M}\boldsymbol{\Sigma_a}\right).
\end{aligned}
\tag{33}
$$

Appendix C - Web scraping code

```
from windmill.authoring import WindmillTestClient
from BeautifulSoup import BeautifulSoup
from time import sleep
from copy import copy
import time
import datetime
import re

def get_massage():
    my_massage = copy(BeautifulSoup.MARKUP_MASSAGE)
    my_massage.append((re.compile(u"document.write(.+);"), lambda
    match: ""))
    my_massage.append((re.compile(u'alt=".+">'), lambda match: ">"))
    return my_massage

def test_scrape_vessel_tracks():
    vessel_infob = []
    vessel_tsb = []

    filename = "VesselData.dat"
    FILE = open(filename,"w")
    FILE.close

    my_massage = get_massage()

    client = WindmillTestClient(__name__)

    while 1 == 1: #Stop when you hit ctrl c
        try:

```

```
            client.open(url="http://www.marinetraffic.com/ais")
31          client.waits.forPageLoad(timeout="60000")
            client.click(id=u"address")
33          client.type(text="english channel", id=u"address")

35          client.waits.forPageLoad(timeout="60000") #6 second
    delay

37          html = client.commands.getPageText()

39          assert html["status"]
            assert html["result"]
41
            soup = BeautifulSoup(html["result"],markupMassage=
    my_massage)
43
            small_vessels = soup.findAll("div", id=u"ship_small")
45          medium_vessels = soup.findAll("div", id=u"ship_medium")
            large_vessels = soup.findAll("div", id=u"ship_large")
47
            vessel_info = []    #Clear vessels in area
49          vessel_ts = []      #Clear vessels time stamps

51          vessel_info,vessel_ts = parse_vessel(small_vessels,
    vessel_info,vessel_ts)
            vessel_info,vessel_ts = parse_vessel(medium_vessels,
    vessel_info,vessel_ts)
53          vessel_info,vessel_ts = parse_vessel(large_vessels,
    vessel_info,vessel_ts)

55          if len(vessel_infob) > 0:
                for x in xrange(0,(len(vessel_infob)-1)):
57                  for y in xrange(0,len(vessel_info)-1):
                        if vessel_infob[x] == vessel_info[y]:
59                          vessel_ts[y] = vessel_tsb[x]    #Replace
     with last AIS value (to enable scraping of data since this last
    record)

61          if len(vessel_info) > 0:
                for x in xrange(0,(len(vessel_info)-1)):
```

```
63                  print "Processing vessel no. ",x," of ",(len(
    vessel_info)-1)
                    try:
65
                        client.open(url="http://marinetraffic.com/
    ais/shipdetails.aspx?mmsi=" + vessel_info[x] + "&header=true")
67                      client.waits.forPageLoad(timeout="60000")

69                      html = client.commands.getPageText()

71                      assert html["status"]
                        assert html["result"]
73
                        soup=BeautifulSoup(html["result"],
    markupMassage=my_massage)
75
                        vessel_details = soup.findAll("div", id=u"
    detailtext")
77
                        text_content = []
79
                        for content in vessel_details:
81                          text_content = content.findAll(text=True
    )

83                      Vesselname = text_content[0]
                        Vesseltype = text_content[6]
85
                        try:
87
                            client.open(url="http://marinetraffic.
    com/ais/datasheet.aspx?MMSI=" + vessel_info[x] + "&TIMESTAMP=1&
    menuid=&datasource=POS&app=&mode=&B1=Search")
89                          client.waits.forPageLoad(timeout=u'60000
    ')
                            html = client.commands.getPageText()
91
                            assert html["status"]
93                          assert html["result"]
```

```
95                              soup = BeautifulSoup(html["result"])
                                table = soup.findAll("table",
    cellpadding = 3)
97
                                for content in table:
99                                  text_content = content.findAll(text=
    True)

101                             MMSI, Speed, LatLon, Course, Timestamp =
    parse_table(text_content)
                                last_timestamp = datetime.datetime.
    strptime(vessel_ts[x],"%Y-%m-%d %H:%M")
103
                                for y in xrange(0,(len(Timestamp)-1)):
105                                 try:
                                        date = Timestamp[y].replace(u'\
    xa0', '')    #Removes \xa0 char ( )
107                                     date = date.encode()
                            #Encode to ascii from unicode

109                                     if datetime.datetime.strptime(
    date,"%Y-%m-%d %H:%M") > datetime.datetime.strptime(vessel_ts[x],
    "%Y-%m-%d %H:%M"):
                                            FILE = open(filename,"a")
111                                         Lat, Lon = LatLon[y].split("
     ")
                                            FILE.write(Timestamp[y] + ",
    " + Lat + "," + Lon + "," + Speed[y] + "," + Course[y] + "," +
    Vesselname + "," + MMSI[y] + "," + Vesseltype + "\n")
113                                         FILE.close

115                                         if datetime.datetime.
    strptime(date,"%Y-%m-%d %H:%M") > last_timestamp:
                                                last_timestamp =
    datetime.datetime.strptime(date,"%Y-%m-%d %H:%M")
117                                 except:
                                        pass
119
                            except:
121                             pass
```

```
123                     except:
                            last_timestamp = datetime.datetime.strptime(
    vessel_ts[x],"%Y-%m-%d %H:%M")
125
                        vessel_ts[x] = last_timestamp.strftime("%Y-%m-%d
    %H:%M")
127
                    vessel_infob = vessel_info      #Store vessel list
129                 vessel_tsb = vessel_ts          #Store AIS time
    stamp

131             #sleep(60) #Delay before next scrape
            except:
133             pass

135 def parse_table(text_content):
        col_count = 1
137     MMSI = []
        Speed = []
139     LatLon = []
        Course = []
141     Timestamp = []

143     for content in text_content:
            if col_count == 1:
145             MMSI.append(content)
            if col_count == 2:
147             Speed.append(content)
            if col_count == 3:
149             LatLon.append(content)
            if col_count == 4:
151             Course.append(content)
            if col_count == 5:
153             Timestamp.append(content)
            col_count += 1
155
            if col_count > 5:
157             col_count = 1
```

```
159     MMSI.remove("MMSI")
        Speed.remove("Speed")
161     LatLon.remove("Latitude / Longitude")
        Course.remove("Course")
163     Timestamp.remove("Timestamp ")

165     return(MMSI, Speed, LatLon, Course, Timestamp)

167 def parse_vessel(vessels, vessel_info, vessel_ts):

169     if len(vessels)>0:

171         for x in vessels:
                try:
173                 mmsi, null, null, null =  x["onclick"].split(",")
                    mmsi = mmsi[15:]
175                 vessel_info.append(mmsi)
                    vessel_ts.append(datetime.datetime.now().strftime("%
    Y-%m-%d %H:%M"))
177             except:
                    pass
179     return (vessel_info, vessel_ts)
```

Appendices/scrape_vessels.py

References

C. Aggarwal and P. Yu. Outlier detection for high dimensional data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Santa Barbara, California, USA*, pages 37–46, 2001.

T. Ahmed, M. Coates, and A. Lakhina. Multivariate online anomaly detection using kernel recursive least squares. In *Proceedings of 26th IEEE International Conference on Computer Communications, Anchorage, Alaska, USA*, pages 625–633, 2007.

M. Augusteijn and B. Folkert. Neural network classification and novelty detection. *International Journal of Remote Sensing*, 23(14):2891–2902, 2002.

Y. Bar-Shalom, X. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. John Wiley & Sons, New York, USA, 2001.

V. Barnett. The study of outliers: purpose and model. *Journal of the Royal Statistical Society*, 27(3):242–250, 1978.

V. Barnett. *Outliers in statistical data*. John Wiley & Sons Ltd., New York, USA, 1994.

R. Basmann. Statistical outlier analysis in litigation support: the case of paul f. engler and cactus feeders, inc., v. oprah winfrey et al. *Journal of Econometrics*, 113(1):159–200, 2003.

M. Basseville. Distance measures for signal processing and pattern recognition. *Signal Processing*, 18(4):349–369, 1989.

C. Bishop. Novelty netection and neural network validation. In *Proceedings of International Conference on Artificial Neural Networks, Amsterdam, The Netherlands*, pages 789–794, 1993.

N. Bomberger, B. Rhodes, M. Seibert, and A. Waxman. Associative learning of vessel motion patterns for maritime situation awareness. In *Proceedings of 9th International Conference on Information Fusion, Florence, Italy*, pages 1–8, 2006.

C. Brax, A. Karlsson, S. Andler, R. Johansson, and L. Niklasson. Evaluating precise and imprecise state-based anomaly detectors for maritime surveillance. In *Proceedings of 13th Conference on Information Fusion (FUSION), Edinburgh, UK*, pages 1–8, 2010.

R. Budzyński, W. Kondracki, and A. Królak. Applications of distance between probability distributions to gravitational wave data analysis. *Classical and Quantum Gravity*, 25(1):015005, 2008.

P. Burridge and A. Taylor. Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis*, 27(5):685–702, 2006.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: a survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009.

P. Chaudhuri. On a geometric notion of quantiles for multivariate data. *Journal of the American Statistical Association*, 91(434):862–872, 1996.

D. Clifton, S. Hugueny, and L. Tarassenko. Novelty detection with multivariate extreme value statistics. *Journal of Signal Processing Systems*, 65:371–389, 2011.

D. Clifton, L. Clifton, S. Hugueny, D. Wong, and L. Tarassenko. An extreme function theory for novelty detection. *Journal of Selected Topics in Signal Processing*, 7:28–37, 2013.

S. Coles. *An introduction to statistical modelling of extreme values*. Springer, London, UK, 2001.

M. Collins. *A protocol graph based anomaly detection system.* PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 2008.

P. Diaconis and D. Freedman. On the consistency of bayes estimates. *The Annals of Statistics*, 44(1):1–26, 1986.

W. Dixon. Analysis of extreme values. *The Annals of Mathematical Statistics*, 21(4):488–506, 1950.

W. DuMouchel. Estimating the stable index α in order to measure tail thickness: a critique. *The Annals of Statistics*, 11(4):1019–1031, 1983.

S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1):36–41, 2007.

S. Franklin, S. Thomas, and M. Brodeur. Robust multivariate outlier detection using mahalanobis distance and modified stahel-donoho estimators. In *Proceedings of The Second International Conference on Establishment Surveys, Buffalo, New York, USA*, 2000.

W. Fung. Critical values for testing in multivariate statistical outliers. *Journal of Statistical Computation and Simulation*, 30(3):195–212, 1988.

J. George, J. Crassidis, T. Singh, and A. Fosbury. Anomaly detection using content-aided target tracking. *Journal of Advances in Information Fusion*, 6 (1):39–56, 2011.

M. Grewal and A. Andrews. *Kalman filtering: theory and practice using MATLAB.* John Wiley & Sons Inc., New Jersey, USA, 2008.

F. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

Y. Gu, Y. Liu, and Zhang Y. A selective KPCA algorithm based on high-order statistics for anomaly detection in hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 5(1):43–47, 2008.

L. Hamel. Model assessment with roc curves. *The Encyclopedia of Data Warehousing and Mining, 2nd Edition, Idea Group Publishers*, pages 1316–1323, 2008.

J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Kittilä, Finland*, pages 379–384, 2010.

H. Huang. *Rank based anomaly detection algorithms.* PhD thesis, Syracuse University, New York, USA, 2013.

N. Imai, S. Terashima, S. Itoh, and A. Ando. 1994 Compilation of analytical data for minor and trace elements in seventeen GSJ geochemical reference samples - igneous rock series. *Geostandards Newsletter*, 19(2):135–213, 1995.

J. Irwin. On a criterion for the rejection of outlying observations. *Biometrika*, 17 (3/4):238–250, 1925.

H. Javitz and A. Valdes. The SRI IDES statistical anomaly detector. In *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, USA*, pages 316–326, 1991.

E. Jaynes. *Probability theory: the logic of science: principles and elementary applications vol 1.* Cambridge University Press, Cambridge, UK, 2003.

R. Lane, D. Nevell, S. Hayward, and T Beaney. Maritime anomaly detection and threat assessment. In *Proceedings of 13th Conference on Information Fusion (FUSION), Edinburgh, UK*, pages 1–8, 2010.

L. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In *Proceedings of 8th International Conference on Machine Learning and Data Mining, Berlin, Germany*, pages 61–75, 2007.

M. Lauer. A mixture approach to novelty detection using training data with outliers. In *Proceedings of 12th European Conference on Machine Learning, Freiburg, Germany*, pages 300–311, 2001.

K. Laws, J. Vesecky, and J. Paduan. Monitoring coastal vessels for environmental applications: application of kalman filtering. In *Proceedings of IEEE/OES 10th Current, Waves and Turbulence Measurements (CWTM), Monterey, California, USA*, pages 39–46, 2011.

R. Laxhammar. Anomaly detection for sea surveillance. In *Proceedings of 11th International Conference on Information Fusion, Cologne, Germany*, pages 1–8, 2008.

R. Laxhammar. *Anomaly detection in trajectory data for surveilance applications.* PhD thesis, Örebro University, Örebro, Sweden, 2011.

R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator. In *Proceedings of 12th International Conference on Information Fusion, Seattle, Washington, USA*, pages 756–763, 2009.

H. Lee and S. Roberts. On-line novelty detection using the kalman filter and extreme value theory. In *Proceedings of 19th International Conference on Pattern Recognition, Tampa, Florida, USA*, pages 1–4, 2008.

X. Li, J. Han, and S. Kim. Motion-alert: automatic anomaly detection in massive moving objects. In *Proceedings of IEEE Intelligence and Security Informatics, San Diego, California, USA*, pages 166–177, 2006.

G. Manson, S. Pierce, K. Worden, T. Monnier, P. Guy, and K. Atherton. Long-term stability of normal condition data for novelty detection. In *Proceedings of SPIE's 7th Annual International Symposium on Smart Structures and Materials, Newport Beach, California, USA*, pages 323–324, 2000.

M. Markou and S. Singh. Novelty detection: a review - part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003a.

M. Markou and S. Singh. Novelty detection: a review - part 2: neural network based approaches. *Signal Processing*, 83(12):2499–2521, 2003b.

S. Mascaro, A. Nicholson, and K. Korb. Anomaly detection in vessel tracks using bayesian networks. In *Proceedings of Eighth UAI Bayesian Modeling Applications Workshop, Barcelona, Spain*, pages 99–107, 2011.

S. Miller, W. Miller, and P. McWhorter. Extremal dynamics: a unifying physical explanation of fractals, 1/f noise, and activated processes. *Journal of Applied Physics*, 73(6):2617–2628, 1992.

G. Münz, L. Sa, and Georg C. Traffic anomaly detection using k-means clustering. In *Proceedings of Zuverlässigkeits-und Verlässlichkeitsbewertung von Kommunikationsnetzen und Verteilten Systemen, Hamburg, Germany*, 2007.

M. Newman. *Networks: an Introduction*. Oxford University Press, Oxford, UK, 2010.

J. P. Nolan. Fitting data and assessing goodness-of-fit with stable distributions. Unpublished Manuscript, 1999.

M. Osborne. *Bayesian gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, University of Oxford, Oxford, UK, 2010.

K. Penny. Appropriate critical values when testing for a single multivariate outlier by using the mahalanobis distance. *Journal of the Royal Statistical Society*, 45 (1):73–81, 1996.

M. Petković, M. Rapaić, Z. Jeličić, and A. Pisano. On-line adaptive clustering for process monitoring and fault detection. *Expert Systems with Applications*, 39(11):1022610235, 2012.

B. Pincombe. Anomaly detection in time series of graphs using ARMA processes. *ASOR Bulletin*, 24(4), 2005.

J. Pinheiro and D. Bates. Unconstrained parameterizations for variance-covariance matrices. *Statistics and Computing*, 6(3):289–296, 1996.

M. Porter, J. Onnela, and P. Mucha. Communities in Networks. *Notices of the american mathematical society*, 56(9):1082–1097, 2009.

I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83(6), 2011.

I. Psorakis, I. Rezek, S. Roberts, and B. Sheldon. Inferring social network structure in ecological systems from spatio-temporal data streams. *Journal of The Royal Society Interface*, 9(76):3055–3066, 2012.

C. Rasmussen and C. Williams. *Gaussian processes for machine learning*. The MIT Press, Massachusetts, USA, 2006.

S. Reece and S. Roberts. The near constant acceleration gaussian process kernel for tracking. *IEEE Signal Processing Letters*, 17(8):707–710, 2010.

H. Ren and Y. Chang. A parallel approach for initialization of high-order statistics anomaly detection in hyperspectral imagery. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium, Honolulu, Hawaii, USA*, pages 1017–1020, 2008.

B. Rhodes, N. Bomberger, M. Seibert, and A. Waxman. Maritime situation monitoring and awareness using learning mechanisms. In *Proceedings of IEEE Military Communications Conference, Atlantic City, New Jersey, USA*, pages 646–652, 2005.

S. Roberts. Extreme value statistics for novelty detection in biomedical signal processing. In *Proceedings of First International Conference on Advances in Medical Signal and Information Processing, University of Bristol, UK*, pages 166–172, 2000.

D. Scott. Outlier detection and clustering by partial mixture modeling. In *Proceedings of Computational Statistics, Prague, Czech Republic*, pages 453–464, 2004.

S. Shapiro and M. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

E. Simpson, S. Roberts, I. Psorakis, and A Smith. Dynamic bayesian combination of multiple imperfect classifiers. In T. Guy, M. Karny, and D. Wolpert, editors, *Decision Making and Imperfection*. Springer, New York, USA, 2013.

S. Verma. Sixteen statistical tests for outlier detection and rejection in evaluation of international geochemical reference materials: example of microgabbro PM-S. *Geostandards Newsletter*, 21(1):5975, 1997.

J. Vesecky, K. Laws, and J. Paduan. Using HF surface wave radar and the ship automatic identification system (AIS) to monitor coastal vessels. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium, University of Cape Town, Cape Town, South Africa*, pages 761–764, 2009.

S. Viaene, R. Derrig, B. Baesens, and G. Dedene. A comparison of state of the art classification techniques for expert automobile insurance claim fraud detection. *Journal of Risk and Insurance*, 69(3):373–421, 2002.

C. Wang, K. Viswanathan, L. Choudur, V. Talwar, W. Satterfield, and K. Schwan. Statistical techniques for online anomaly detection in data centers. In *Proceedings of 12th IFIP/IEEE International Symposium on Integrated Network Management, Trinity College Dublin, Ireland*, pages 385–392, 2011.

M. Welling. Robust higher order statistics. In *Proceedings of Tenth International Workshop on Artificial Intelligence and Statistics, The Savannah Hotel, Barbados*, pages 405–412, 2005.

J. Will, L. Peel, and C. Claxton. Fast maritime anomaly detection using kd-tree gaussian processes. In *Proceedings of 2nd IMA Conference on Mathematics in Defence, Defence Academy of the United Kingdom, Swindon*, 2011.

S. Yonga, D. Jeremiah, and M. Purvisa. Novelty detection in wildlife scenes through semantic context modelling. *Pattern Recognition*, 45(9):34393450, 2012.

M. Zandipour, B. Rhodes, and N. Bomberger. Probabilistic prediction of vessel motion at multiple spatial scales for maritime situation awareness. In *Proceed-*

ings of 11th International Conference on Information Fusion, Cologne, Germany, pages 1–6, 2008.

T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada*, pages 103–114, 1996.