

Applying Gaussian Processes in the Sport of Cricket

Louie Middle

Bachelor of Science in Computer Science
The University of Bath
2023

This dissertation may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Applying Gaussian Processes in the Sport of Cricket

Submitted by: Louie Middle

Copyright

Attention is drawn to the fact that copyright of this dissertation rests with its author. The Intellectual Property Rights of the products produced as part of the project belong to the author unless otherwise specified below, in accordance with the University of Bath's policy on intellectual property (see https://www.bath.ac.uk/publications/university-ordinances/attachments/Ordinances_1_October_2020.pdf).

This copy of the dissertation has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the dissertation and no information derived from it may be published without the prior written consent of the author.

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

〈 The abstract should appear here. An abstract is a short paragraph describing the aims of the project, what was achieved and what contributions it has made. 〉

Contents

Acknowledgements	vii
1 Introduction	1
1.1 Moneyball	1
1.2 Indian Premier League	2
1.3 Project Plan	2
1.4 Resources Required	3
2 Literature and Technology Survey	4
2.1 Existing Machine Learning in Sport	4
2.2 Existing Machine Learning in Cricket	5
2.2.1 Existing Machine Learning in the Indian Premier League	5
2.3 Research with similar goals	5
2.4 Drawbacks of Existing Methods	6
2.5 Gaussian Processes in Sport	6
3 Background	7
3.1 Gaussian Processes	7
3.1.1 Gaussian Processes for Regression	8
3.1.2 Kernels	9
3.1.3 Predictions and Posterior	10
3.1.4 Gaussian Processes for Classification	12
3.1.5 Sparse Gaussian Processes	12
3.2 Data Association with Gaussian Processes	12
3.2.1 Data Association	12
3.3 Gradient Boosting	12
4 Method	13
5 Results	14
6 Discussion	15
7 Conclusion	16
8 Future Work	17
9 Personal Experiences	18
9.1 Short Section Title	18

9.2	Example Lists	18
9.2.1	Enumerated	18
9.2.2	Itemised	18
9.2.3	Description	19
Bibliography		20
A Mathematical Background		22
A.1	Cholesky Factorisation	22
B Design Diagrams		23
C Raw Results Output		24
D Code		25
D.1	File: yourCodeFile.java	26

List of Figures

1.1	Gantt Chart For Project	3
2.1	Number of papers using a particular ML algorithm group (Horvat and Job, 2020)	4
3.1	SE kernel with $\sigma = 1$ and $l = 1$ (Kaiser, 2017)	10
3.2	SE kernel with $\sigma = \sqrt{2}$ and $l = 1$ (Kaiser, 2017)	10
3.3	SE kernel with $\sigma = 1$ and $l = 0.25$ (Kaiser, 2017)	10
3.4	A data association problem consisting of two generating processes, one of which is a signal we wish to recover and one is an uncorrelated noise process (Kaiser et al., 2018)	12

List of Tables

9.1	An example table	18
-----	----------------------------	----

List of Algorithms

1	Algorithm for Gaussian process regression	11
---	---	----

Acknowledgements

I would like to acknowledge my supervisor Adam Hartshorne. I would like to acknowledge my partner Hoi Ching Leung. I would like to acknowledge my parents.

Chapter 1

Introduction

Over the last few decades the amount of data driven techniques to improve the outcomes of sports games has increased greatly. The multi-billion pound market of cricket is no exception. There is a strong incentive to improve the techniques used to better the results of teams. This study aims to investigate the possibility of predicting the outcome of a bowler batter match-up in cricket using modern machine learning techniques and how this could then aid cricket bowling choices and team selection. The main training and testing data will be the 2022 Indian Premier League (IPL) season (TODO: Change) The hope is that using knowledge of bowler pitch trajectories and batter shot trajectories gathered from modern ball tracking will add another layer of granularity in addition to simply considering the resultant runs scored of each delivery. Furthermore, building a model which can incorporate pitch, atmospheric and ground conditions could further improve any models.

1.1 Moneyball

The release of Moneyball (Lewis, 2004), was a big driver in the increased use of statistical driven techniques in Baseball player selection and scouting techniques. Pioneered by the likes of Bill James and Sabermetrics, Moneyball has entered baseball's lexicon; teams that value Sabermetrics are often said to be playing "Moneyball". One of the notable benefits of a Moneyball approach is in reducing player salaries, whilst maintaining high performance. Notable recent Moneyball successes include the Tampa Bays, whose entire 2019 roster was around 63% of the total budget of the \$40 million the Houston Astros were spending on Gerrit Cole's contract. It was reported that the Rays spending totalled \$648,000 per victory, compared to the Astros \$1.54 million per win. Despite this large difference pay, the Rays still had a successful season and finished second in the American League East (Sports, 2019).

Similar successes can be found in other sports, such as association football (or soccer). In 2010 Liverpool F.C. were purchased by Boston Red Sox owner John W. Henry. With the Red Sox, Henry hired Sabermetrics pioneer Bill James and their Moneyball approach saw the team win the World Series in 2004, 2007, 2013 and 2018. With Liverpool, Henry hired University of Cambridge PhD Ian Graham in 2012 as head of analysis and Jürgen Klopp as Manager in 2015 (News, 2022). Graham influenced the signings of key players, such as Mohammed Salah, Philippe Coutinho and Naby Keïta. Grahams data suggested Salah would pair especially well with Roberto Firmino, who creates more expected goals than nearly anyone else in his position (Echo, 2019). Expected goals turned to real goals in the 2017-2018 season, with Salah scoring

32 goals and Firmino scoring 15. The combination of Klopp and his intuitive knowledge, mixed with the likes of Grahams data-driven knowledge, has led to Liverpool having fantastic recent success winning the 2018-2019 UEFA Champions League and the 2019-2020 English Premier League.

1.2 Indian Premier League

The Indian Premier League (IPL) is a professional cricket league based on the Twenty20 format. As reported by ESPNcricinfo (2018), Star Sports invested \$2.55 billion for exclusive broadcasting rights for the 2018 IPL season. This season saw a 29% increment in the number of viewers, through both digital streaming and television. The interest in the IPL is clear to see, thus increasing the desire to use modern techniques to improve results.

1.3 Project Plan

TODO: This can be removed

There are 26 weeks from Friday the 4th November until the final deadline of Friday the 5th May. The individual project is 24 credits out of a total 60 credits for the year, meaning 40% of my time can be used for the project. This is 10.4 weeks. To allow for buffer and holidays I will assume I have 8 working weeks to complete the project. I have split my project into 3 main sections:

1. Literature review and pre-processing (2 weeks)
2. Developing and improving models (4 weeks)
3. Analysis of models and write up (2 weeks)

The buffer time can be used for any road bumps, or sections that need it. See Gantt chart for overview of plan in figure 1.1.

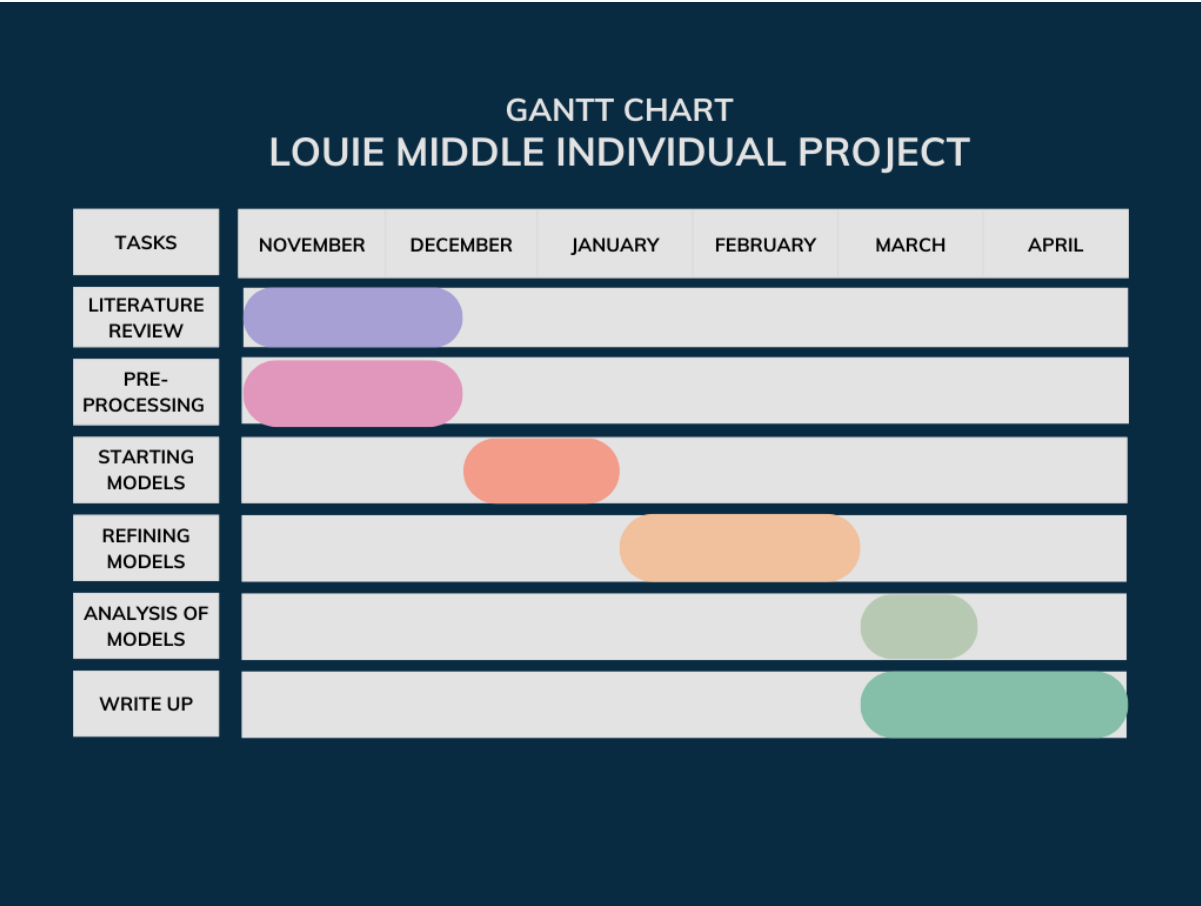


Figure 1.1: Gantt Chart For Project

1.4 Resources Required

TODO: This section should be removed or amended. I think get some data from just one year, or all years and compare to test Horvat and Job.

In order to train any machine learning model an appropriate amount of data for training and testing. Because it is not beneficial to use data that is too old (Horvat and Job, 2020), a recent season’s worth of data would be good, saving some data for testing.

Training machine learning models may also require appropriate computing power depending on the models used and size of the data set. This could potentially be achieved with the University of Bath’s Hex cluster.

Chapter 2

Literature and Technology Survey

2.1 Existing Machine Learning in Sport

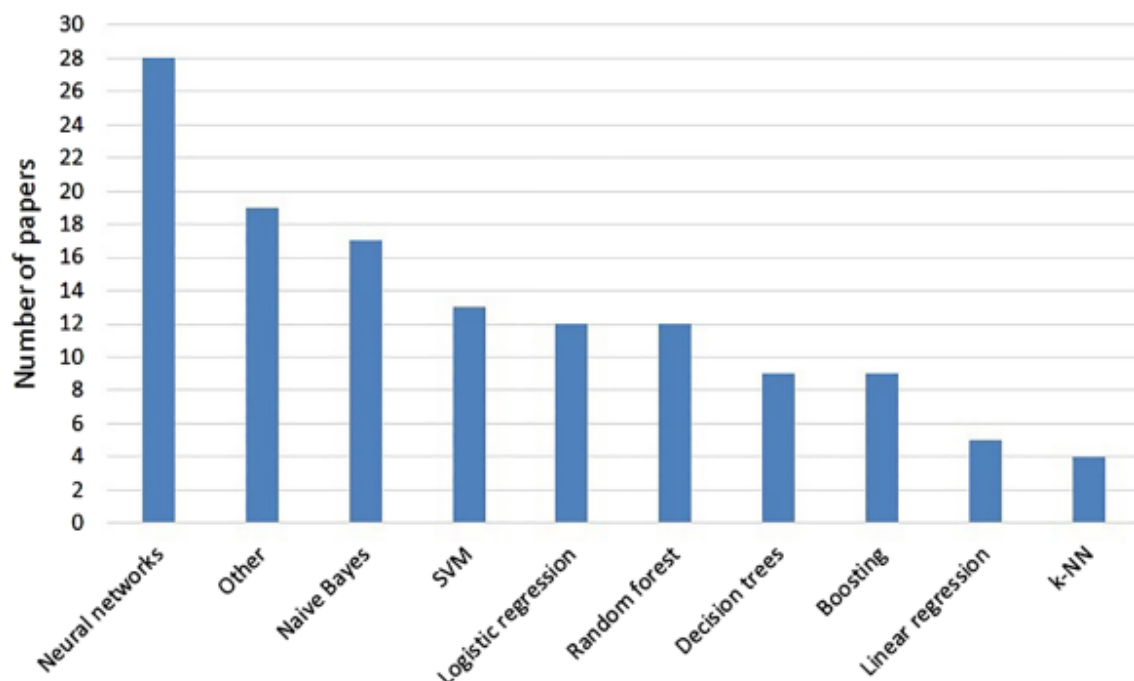


Figure 2.1: Number of papers using a particular ML algorithm group (Horvat and Job, 2020)

TODO: Could be worth putting the one with publication year in to show uptick in research? But not necessary right now.

Figure 2.1 from Horvat and Job (2020)'s literature review of machine learning in sport for score prediction shows what current methods are currently used in the field as of May 2020. As you can see, neural networks have a large research output, followed by other popular methods such as regression, SVMs, regression, decision trees, and gradient boosting with k-NN being the least used. It is important to note, that just because a large body of research has been carried out with certain methods, it does not necessarily mean they are the best.

Horvat and Job (2020) also showed that including too many seasons worth of data for training models reduces the quality of results. To those with a basic knowledge of cricket and sport this is not surprising given that in just a few years, many things related to team composition and tactics can change. The best prediction results were achieved by researchers who used data from a single season and a data segmentation evaluation method. When using data from a single season, most of the data is used for training and a small portion for testing. Some researchers used the same data set for training and testing yielding unrealistically accurate results.

2.2 Existing Machine Learning in Cricket

Kampakis and Thomas (2015) used Naïve Bayes, logistic regression, random forests and gradient boosted decision trees to predict the outcome of English County 20 over Cricket Matches from 2009 - 2014. The performance of each algorithm was assessed using one year's data as the training data set and the following year's data for testing. Each model was tested over these seasons and achieved an accuracy of 62.4% for Naïve Bayes, 60.1% for logistic regression, 55.6% for random forests and 57.2% for decision trees.

2.2.1 Existing Machine Learning in the Indian Premier League

Saikia, Bhattacharjee and Lemmer (2012) have used Artificial Neural Network models to predict the performance of bowlers based on their performance in the first three seasons of the IPL. When the predicted results were validated with the players performances in the fourth season, the model had an accuracy of 71.43%.

2.3 Research with similar goals

The Singularity-PA model (Silver and Huffman, 2021) wanted to attempt to solve one of the most fundamental questions in baseball: How can we predict the outcome of a batter vs. pitcher plate appearance (PA)? This is similar to my goal with cricket: To predict the outcome of batter vs bowler for an over.

The Singularity-PA model (Silver and Huffman, 2021) was able to accurately predict the results of a batter versus pitcher plate appearance using a neural-network based AI model. The details of the model used are vague, however the network was able to take in 87 inputs and then output probabilities for each of the 21 possible outcomes of a plate appearance (PA) in baseball. Comparisons can be made between this and cricket. A plate appearance can be compared to an over, comprising 6 balls (or more including no balls and wides) between a single bowler and 1 or more batsmen. The outcome of the over could be considered to be the runs scored.

Silver and Huffman (2021) also split their player base up by how many PAs they had for each player. The best players had greater than 500 PAs worth of data each, but the vast majority had less than 100. SingularityPA was able to accurately predict the result of match-ups for these players with fewer PAs better than existing solutions. Parallels can be drawn between this and cricket, as there are often players who have little data, yet team selectors would want to know who the best player is to match-up against them.

Extending Singularity-PA with Markov chains improved more complicated strategies, such as

optimal player lineups or to decide on pinch hitters and relief pitchers. Similarly, in cricket the batting lineup and choice of bowler at different points in a game have a large impact on the score. In the example provided, Singularity-PA's predicted runs scored for an optimal lineup was 6.7% better than the actual lineup in the 2019 National League All-Star game. It is important not to compare baseball and cricket too closely, but the techniques used by Silver and Huffman could potentially work well in Cricket.

2.4 Drawbacks of Existing Methods

Sport outcome predictors are most commonly used by supervised ML methods, typically classification methods or regression methods (Horvat and Job, 2020). Whilst existing research can achieve impressive results, surprises in sport still happen. For example, the odds of Leicester City winning the Premier League in the 2015/2016 season were 1-5000. However, analysis of their performances show their title was absolutely deserved.

This could be explained in that a problem with existing models is that they only output a single value. There is no uncertainty in the output as to how confident the model is in its prediction. This has a problem in that making decisions based on this prediction becomes much more difficult, as decision makers can't be sure how much to trust the prediction. Furthermore, certain inputs exist where very little similar training data might exist. In such cases, the uncertainty in a models output should be much greater. Yet, existing models will still provide a prediction the same as it would for inputs where there was a large amount of training data.

TODO: talk about (Blumberg, 2020).

2.5 Gaussian Processes in Sport

TODO: Talk about Gaussian processes in sport

Chapter 3

Background

In this chapter I will cover the requisite background for Gaussian processes and gradient boosting, the two key methods used in this project.

3.1 Gaussian Processes

(Griffiths, 2023)

(Rasmussen and Williams, 2006)

(Kaiser, 2017)

(Yi, 2019) TODO: This has good explanations of stuff

TODO: Put cholesky decomposition and KL divergence in the appendix

This section introduces Gaussian processes and the following section then explains how to extend their application to solve data association problems. This project is mostly concerned with using Gaussian processes for classification, but a background on using Gaussian processes for regression is provided first as a basis before their extension to classification tasks. As Gaussian processes are not computationally cheap sparse approximations were used in this project and are also reviewed in this section.

Gaussian processes are a non-parametric regression model. *Non-parametric models* are not based on insights about the concrete structure of the function to be modelled, but instead make assumptions about the function itself, such as its smoothness or differentiability. Instead of modeling a distribution of parameter values, a non-parametric model tries to find a distribution $p(f^*)$ of probable functions that represents the function f to be estimated.

In addition, Gaussian processes are a supervised learning technique, where we start with a training data set \mathcal{D} of n observations, $\mathcal{D} = (\mathbf{x}_i, y_i | i = 1, \dots, n)$ where \mathbf{x} denotes an input vector (covariates) of dimension D and y denotes an output or target. The column vector inputs for all n cases are aggregated in the $D \times n$ design matrix X and the targets collected in the vector \mathbf{y} , such that $\mathcal{D} = (X, \mathbf{y})$.

3.1.1 Gaussian Processes for Regression

Weight Space View

TODO: This section might not be necessary, but might be good to show understanding....

One can think of a Gaussian process as defining a distribution over functions and inference taking place directly in the space of functions. Although this view is appealing, it is difficult to grasp on first attempt, and so we will start with reviewing the *weight-space view*.

First, let's review the standard linear regression model with Gaussian noise

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}, \quad y = f(\mathbf{x}) + \epsilon \quad (3.1)$$

where \mathbf{x} is the input vector, \mathbf{w} is a vector of weights of the linear model, f is the function value, and y is the observed target value. It is often assumed that the observed values differ from the function values by some noise, which we will treat as an independent, identically distributed Gaussian distribution, with zero mean and variance σ_n^2 .

$$\epsilon \sim \mathcal{N}(0, \sigma_n^2) \quad (3.2)$$

This noise assumption together with the model gives rise to the likelihood, the probability density of the observations given the parameters, which is factored over cases in the training set to give

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^n p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - \mathbf{x}_i^T \mathbf{w})^2}{2\sigma_n^2}\right) \quad \text{TODO : Candothertelater} \quad (3.3)$$

We put a zero mean Gaussian prior with covariance matrix Σ_p on the weights

$$\mathbf{w} \sim \mathcal{N}(0, \Sigma_p) \quad (3.4)$$

Inference is based on the posterior distribution over the weights computed by Baye's rule, given by

$$P(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \quad (3.5)$$

Note that prior $p(\mathbf{w})$ neglects the conditioning on \mathbf{X} , as it is independent of the inputs. The normalising constant $p(\mathbf{y}|\mathbf{X})$, also known as the marginal likelihood is independent of the weights and is given by

$$P(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w} \quad (3.6)$$

Since the number of observations is finite and the function f lives in an infinite dimensional function space, the estimation of f is uncertain and based on prior assumptions about its structure.

TODO: Finish this later when you can ask Adam about it....

To make predictions we average over all possible parameter values, weighted by their posterior probability. Thus the predictive distribution for $f_* \triangleq f(\mathbf{x}_*)$ at \mathbf{x}_* is given by averaging the output of all possible linear models w.r.t the Gaussian posterior

$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|X, \mathbf{y})d\mathbf{w} \\ &= \mathcal{N}\left(\frac{1}{\sigma_n^2}\mathbf{x}_*^T A^{-1}X\mathbf{y}, \mathbf{x}_*^T A^{-1}\mathbf{x}_*\right). \end{aligned} \quad (3.7)$$

The predictive distribution is again Gaussian.

Function-space View

Definition 3.1.1 (Gaussian Process). A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

Gaussian processes are a generalisation of the Gaussian distribution to function spaces. A multivariate Gaussian $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ describes a distribution over the finitely many elements in the vector \mathbf{x} . Every element of \mathbf{x} is normally distributed. For two points in \mathbf{x} , x_i and x_j , their covariance is given by $\text{cov}[x_i, x_j] = \Sigma_{ij}$.

A Gaussian process is completely specified by its mean function, $m(\mathbf{x})$, and its co-variance function $k(\mathbf{x}, \mathbf{x}')$. Usually the mean function is assumed to be constant zero, but this need not be the case. A mean function and covariance function can be defined for a real process $f(x)$ as

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \end{aligned} \quad (3.8)$$

where we will write the Gaussian process as

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (3.9)$$

The covariance function, also called a kernel, specifies the covariance between pairs of random variables. It is the covariance functions which encode the assumptions about the underlying function (see section 3.1.2).

3.1.2 Kernels

Kernels are crucial in encoding the assumptions about the function a Gaussian process should estimate. It is a measure of similarity of different points in the observed data and of new points to be predicted. For example, a natural assumption is to assume the closer two points lie, the more similar their function values should be. Furthermore, when predicting test points, training points close to it are probably more informative than those further away. However, it should be noted, that this need not be the case. For example, consider a sinusoidal wave where two points which are multiple wavelengths apart should have similar function values. A kernel that only depends on the distance between two points is called *stationary*. Conversely, kernels that do depend on two points position in the input space are called *non-stationary*.

A common kernel and one used throughout this project is the squared exponential (SE) covariance function (also known as the radial basis function, RBF). This is defined by

$$\text{cov}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}|\mathbf{x}_p - \mathbf{x}_q|^2\right). \quad (3.10)$$

Kernels have characteristic length scales which informally can be thought of as roughly the distance you have to move in input space before the function value will change significantly. For eq 3.10 the length-scale is one. By replacing $\mathbf{x}_p - \mathbf{x}_q$ with $\mathbf{x}_p - \mathbf{x}_q/l$ for some positive constant l we can change the characteristic length-scale of the process. Choice of such parameters will be discussed more later.

TODO: Explain σ !!!

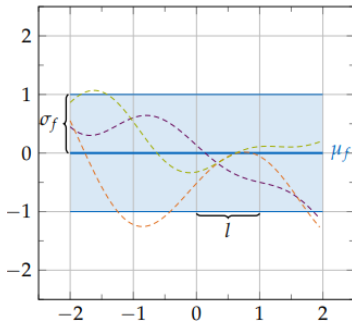


Figure 3.1: SE kernel with $\sigma = 1$ and $l = 1$ (Kaiser, 2017)

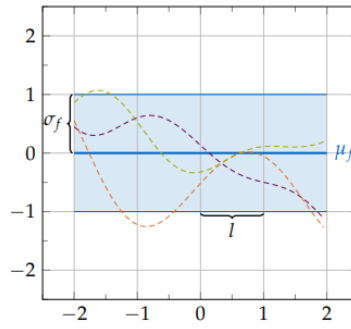


Figure 3.2: SE kernel with $\sigma = \sqrt{2}$ and $l = 1$ (Kaiser, 2017)

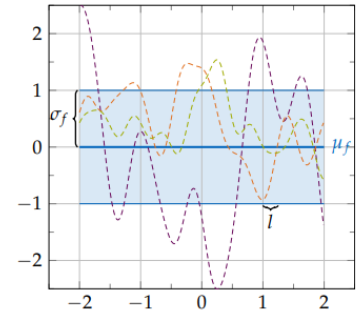


Figure 3.3: SE kernel with $\sigma = 1$ and $l = 0.25$ (Kaiser, 2017)

Figure 3.1, figure 3.2 and figure 3.3 compare sample functions drawn from Gaussian processes with SE kernels with different hyperparameters. Since the mean function $m(x)$ is assumed to be constant zero, the kernel specifies the prior assumptions about the function. The SE kernel describes arbitrary smooth functions. The hyperparameters l and σ of the kernel describe the dynamic range in the x and y directions respectively.

3.1.3 Predictions and Posterior

In order to use Gaussian processes for regression, it is necessary to combine observations with a Gaussian process prior $f \sim GP(0, K)$. The distribution is obtained by integrating over all possible latent function values f and therefore taking all possible functions into account. This is called the *marginalisation of f* .

The joint distribution of the training outputs, \mathbf{y} , and the test outputs \mathbf{f}_* according to the prior is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \quad (3.11)$$

If there are n training points and n_* test points then $K(X, X_*)$ denotes the $n \times n_*$ matrix of covariances evaluated at all pairs of training and test points, and similarly for $K(X, X)$, $K(X_*, X)$ and $K(X_*, X_*)$.

Using this we arrive at the key predictive equations for Gaussian process regression

$$\mathbf{f}_* | X, \mathbf{y}, X_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)) \quad (3.12)$$

$$\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_* | X, \mathbf{y}, X_*] = K(X, X_*)[K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y} \quad (3.13)$$

$$\text{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \quad (3.14)$$

Eq. 3.13 and eq. 3.14 represent the mean function and the covariance function of the Gaussian posterior process respectively.

From here forwards, a compact form of $K(X, X)$ and $K(X_*, X_*)$ etc. will be introduced where $K = K(X, X)$ and $K_* = K(X_*, X_*)$.

Computing $[K + \sigma_n^2 I]^{-1}$ costs $O(N^3)$, but can be done as a preprocessing step since it is independent of the test points. After this, each single test point costs $O(N)$. To predict its variance it is still necessary to perform matrix multiplication which costs $O(N^2)$.

Marginal Likelihood

The *marginal likelihood* is the marginalisation over the function values \mathbf{f} . By observing that $\mathbf{y} \sim \mathcal{N}(0, K + \sigma_n^2 I)$ yields the log marginal likelihood

$$\log p(\mathbf{y} | X) = -\frac{1}{2} \mathbf{y}^T (K + \sigma_n^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{n}{2} \log 2\pi. \quad (3.15)$$

With this, we can create an algorithm for Gaussian process regression, which is given in alg. 1. The matrix inversion required by eq. 3.13 and 3.14 uses Cholesky factorisation, explained in section A.1.

Algorithm 1 Algorithm for Gaussian process regression

Require: X (inputs), \mathbf{y} (targets), k (covariance function), σ_n^2 (noise level), X_* (test input)

- 1: $L \leftarrow \text{cholesky}(K + \sigma_n^2 I)$
 - 2: $\alpha \leftarrow L^T \setminus (L \setminus \mathbf{y})$ ▷ eq. 3.13
 - 3: $\bar{\mathbf{f}}_* \leftarrow \mathbf{k}_*^T \alpha$ ▷ eq. 3.13
 - 4: $\mathbf{v} \leftarrow L \setminus \mathbf{k}_*$ ▷ eq. 3.14
 - 5: $\mathbb{V}[\mathbf{f}_*] \leftarrow k(X_*, X_*) - \mathbf{v}^T \mathbf{v}$ ▷ eq. 3.14
 - 6: $\log p(\mathbf{y} | X) \leftarrow -\frac{1}{2} \mathbf{y}^T \alpha - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$ ▷ eq. 3.15
 - 7: **return:** $\bar{\mathbf{f}}_*$ (mean), $\mathbb{V}[\mathbf{f}_*]$ (variance), $\log p(\mathbf{y} | X)$ (log marginal likelihood)
-

Loss Function

For practical applications, we are forced to make a decision how to act - we need a point prediction. To achieve this, we need a *loss function* $\mathcal{L}(y_{\text{true}}, y_{\text{guess}})$ which specifies the loss incurred by guessing the value y_{guess} when the true value is y_{true} .

TODO: Loss functions...

3.1.4 Gaussian Processes for Classification

A Gaussian process is a generalisation of the gaussian probability distribution. Both classification and regression can be seen as function approximation problems. Unfortunately, the solution of classification problems using Gaussian processes is tougher than regression problems. For regression problems, the likelihood is often assumed to be Gaussian. A Gaussian process prior combined with a Gaussian likelihood gives a posterior Gaussian process over functions, where everything remains analytically tractable. For classification models, the Gaussian likelihood is inappropriate; a different likelihood such as a Bernoulli likelihood must be used.

3.1.5 Sparse Gaussian Processes

TODO: Write about sparse GPs

3.2 Data Association with Gaussian Processes

3.2.1 Data Association

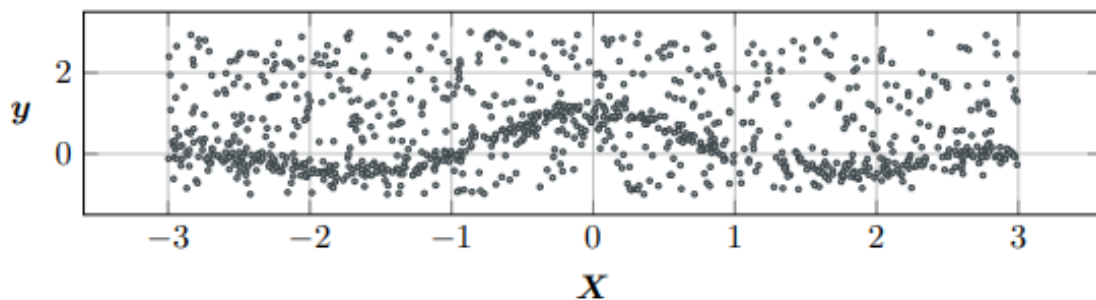


Figure 3.4: A data association problem consisting of two generating processes, one of which is a signal we wish to recover and one is an uncorrelated noise process (Kaiser et al., 2018)

A *data association problem* is one where we consider the data to have been generated by a mixture of processes and we are interested in factorising the data into these components. For example, as described by Kaiser et al. (2018), figure 3.4 could represent faulty sensor data, where sensor readings are disturbed by uncorrelated and asymmetric noise. Standard machine learning approaches can pollute any models, where the model starts to explain the noise instead of the underlying signal.

The cricket dataset could be deemed a data association problem....

(Kaiser et al., 2018) (Liu et al., 2020)

3.3 Gradient Boosting

TODO: Describe XGBoost

Chapter 4

Method

Chapter 5

Results

Chapter 6

Discussion

Chapter 7

Conclusion

Code can be output inline using `\lstinline|some code|`. For example, this code is inline: **public static int** example = 0; (we have used the character `|` as a delimiter, but any non-reserved character not in the code text can be used.)

Code snippets can be output using the `\begin{lstlisting} ... \end{lstlisting}` environment with the code given in the environment. For example, consider listing 7.1, below.

Listing 7.1: Example code

```
public static void main() {  
  
    System.out.println("Hello World");  
  
}
```

Code listings are produced using the package 'listings'. This has many useful options, so have a look at the package documentation for further ideas.

Chapter 8

Future Work

Chapter 9

Personal Experiences

9.1 Another Section With a Long Title and Whose Title Is Abbreviated in the Table of Contents

Table 9.1: An example table

Items	Values
Item 1	Value 1
Item 2	Value 2

Another section, just for good measure. You can reference a table, figure or equation using `\ref`, just like this reference to Table 9.1.

9.2 Example Lists

9.2.1 Enumerated

1. Example enumerated list:
 - a nested enumerated list item;
 - and another one.
2. Second item in the list.

9.2.2 Itemised

- Example itemised list.
 - A nested itemised list item.
- Second item in the list.

9.2.3 Description

Item 1 First item in the list.

Item 2 Second item in the list.

Bibliography

- Blumberg, R., 2020. A fully transparent expected-goals model [Online]. Available from: <https://robbblumberg.github.io/exp-goals/>.
- Echo, L., 2019. The unknown liverpool backroom analyst who convinced juergen klopp to commission transfers of salah and keita [Online]. Available from: <https://www.liverpoolecho.co.uk/sport/football/transfer-news/unknown-liverpool-backroom-analyst-who-16320524>.
- ESPNcricinfo, 2018. How can the ipl become a global sports giant? [Online]. Available from: <https://www.espnecricinfo.com/story/tim-wigmore-how-can-the-ipl-become-a-global-sports-giant-1150459>.
- Griffiths, R.R., 2023. *Applications of gaussian processes at extreme lengthscales: From molecules to black holes* [Online]. Ph.D. thesis. Available from: <https://arxiv.org/pdf/2303.14291.pdf>.
- Horvat, T. and Job, J., 2020. The use of machine learning in sport outcome prediction: A review. *Wiley interdisciplinary reviews: Data mining and knowledge discovery* [Online], 10(5), p.e1380. Available from: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1380?casa_token=aXIDtR-gOHQAAAAA%3A-YGsS9bhdi0kYkurvCmBCdZpbAx-MV4VF806rtp_h1CXJlCC0pg4hFFknpmc9dAb3wBAVu7NZ_F0BQ.
- Kaiser, M., 2017. *Incorporating uncertainty into reinforcement learning through gaussian processes* [Online]. Master's thesis. Available from: <https://github.com/mrksr/masters-thesis/blob/master/thesis/thesis.pdf>.
- Kaiser, M., Otte, C., Runkler, T. and Ek, C.H., 2018. Data association with gaussian processes [Online]. Available from: <https://doi.org/10.48550/ARXIV.1810.07158>.
- Kampakis, S. and Thomas, W., 2015. Using machine learning to predict the outcome of english county twenty over cricket matches [Online]. Available from: <https://arxiv.org/ftp/arxiv/papers/1511/1511.05837.pdf>.
- Lewis, M., 2004. *Moneyball: The art of winning an unfair game*. WW Norton & Company.
- Liu, H., Ong, Y.S., Jiang, X. and Wang, X., 2020. Modulating scalable gaussian processes for expressive statistical learning [Online]. Available from: <https://doi.org/10.48550/ARXIV.2008.12922>.
- News, L.F., 2022. Who is ian graham? liverpool suffer second transfer blow after julian ward departure [Online]. Available from: <https://www.liverpool.com/liverpool-fc-news/features/ian-graham-liverpool-transfer-ward-25596815>.

- Rasmussen, C.E. and Williams, C.K., 2006. *Gaussian process for machine learning* [Online], vol. 2. MIT Press Cambridge, MA. Available from: <http://gaussianprocess.org/gpml/>.
- Saikia, H., Bhattacharjee, D. and Lemmer, H.H., 2012. Predicting the performance of bowlers in ipl: An application of artificial neural network. *International journal of performance analysis in sport*, 12(1), pp.75–90.
- Silver, J. and Huffman, T., 2021. Baseball predictions and strategies using explainable ai [Online]. *The 15th annual mit sloan sports analytics conference*. Available from: <https://singularity.com/static/assets/sloan-singularity.pdf>.
- Sports, F., 2019. The tampa bays are the undisputed kings of moneyball [Online]. Available from: <https://www.foxsports.com/stories/mlb/the-tampa-bay-rays-are-the-undisputed-kings-of-moneyball>.
- Yi, W., 2019. Understanding gaussian process, the socratic way [Online]. Available from: <https://towardsdatascience.com/understanding-gaussian-process-the-socratic-way-ba02369d804>.

Appendix A

Mathematical Background

A.1 Cholesky Factorisation

Appendix B

Design Diagrams

Appendix C

Raw Results Output

Appendix D

Code

D.1 File: yourCodeFile.java

```
// This is an example java code file , just for illustration  
purposes  
public static void main() {
```

```
    System.out.print ("Hello World");  
}
```