Cricket Delivery Outcome Prediction using XGBoost and Mixtures of Gaussian Processes: A Comparative Study of Model Performance

Louie Middle

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.

Cricket Delivery Outcome Prediction using XGBoost and Mixtures of Gaussian Processes: A Comparative Study of Model Performance

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

List of Figures

List of Tables

List of Algorithms

Acknowledgements

I would like to acknowledge my supervisor Adam Hartshorne for his support and advice throughout the development of this project. In addition, my partner Hoi Ching Leung has provided much personal support for me throughout this challenging final year. Lastly, I would like to thank my parents, for who without I would not have had the amazing oppurtunities in life that I have had, including University and the chance to work on this project.

Glossary

TODO: Write a glossary of terms

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.

Cricket is a game of fine margins; matches can be decided by a small number of runs. Therefore, every insight that can be made as to what is good or bad cricket is extremely valuable. For this reason, there is a strong incentive to improve the machine learning techniques used in current sports analytics to improve a teams performance.

This study aims to investigate the possibility of predicting the outcome of a cricket delivery using modern machine learning techniques and how this could then aid cricket bowling choices and team selection. The main training and testing data used will be the last 10 years of the Indian Premier League (IPL). 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 off each delivery. Furthermore, building a model which can incorporate pitch, atmospheric and ground conditions could further improve any models.

1.1 Sports Analytics

Sports analytics is the use of data analysis techniques to gain insights and make decisions related to various aspects of sports, such as player performance, team strategy, and game outcomes. It involves the use of statistical and machine learning techniques to extract meaningful information from large and complex datasets.

Sports analytics has become an integral part of the decision-making process for many professional teams and organisations. By analyzing data on player performance and team strategies, coaches and managers can make more informed decisions on player recruitment, game tactics, and training regimes.

1.2 Drawbacks of Current Methods in Sports Analytics

Sport predictors are most commonly using supervised ML methods, typically Neural networks, classification methods or regression methods [Horvat and Job, 2020] (see sections 2.3 and 2.4). Whilst existing research can achieve impressive results, unpredictable outcomes 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. Examples like this illustrate the limitations of current predictions in sport.

Currently in sports analytics and machine learning in general there is a go to set of methods that are very popular. Particularly neural networks and boosting algorithms are currently popular and have a large research output. Whilst they have gained popularity for their ability to solve complex problems and deliver accurate predictions, all methods have drawbacks.

1.2.1 Overfitting

One significant drawback of neural networks and boosting models is that they can easily overfit the training data, meaning they memorize the data and fail to generalize well to new data. Regularization techniques such as L1 or L2 regularization, dropout, or early stopping can help reduce overfitting, but these methods may not always be sufficient.

1.2.2 Discretisation of Continuous Features

TODO: Talk about how they do this in football.

Gradient boosting and neural networks, along with other machine learning techniques such as decision trees and random forests, discretize their input space by binning continuous features into intervals or categories. This allows these models to handle continuous variables as categorical variables, simplifying the modeling process. However this is another potential drawback, as this discretization can lead to a loss of information, as the fine-grained details of the continuous variables are lost. This can be particularly problematic for variables with high granularity or subtle variations, where the loss of information can result in reduced accuracy.

This is the case in cricket, where for example, a perfect yorker [1] is a very difficult ball to play against. However, becoming fuller or shorter can turn the ball into a full toss [2] or half volley [3] respectively, which are both considered easier deliveries to score high runs off of. So in the case of binning a feature such as the location of pitch bounces, it wouldn't appropriately represent the realities of this continuous feature.

In addition to discretization, another way to handle continuous variables is by bucketing data into tabular format. This approach involves partitioning continuous variables into distinct ranges or "buckets" and assigning them categorical labels. This method preserves more of the original information compared to discretization, and is particularly useful for variables with high granularity. Bucketing can be applied to different types of variables, including numerical and time-series data, and is commonly used in machine learning models such as decision trees, random forests, and gradient boosting.

However, bucketing can also be prone to certain issues such as overfitting and bias, particularly when the number of buckets is too large or too small. Furthermore, the choice of bucketing method and the number of buckets can be subjective, and may require experimentation and tuning. Overall, both discretization and bucketing have their advantages and disadvantages, and the choice of method depends on the specific data and modeling needs.

---

[3]In cricket, a yorker is a delivery which hits the cricket pitch around the batsman's feet.
[3]In cricket, a full toss is a delivery which does not pitch before it reaches the batsman.
[3]In cricket, a half volley is a delivery which hits the cricket pitch ahead of a good length, but before the batsman's feet.

1.2.3 Limited Data

Limited data is another challenge faced by machine learning models. In some cases, obtaining enough data to train a model can be expensive, time-consuming, or simply impossible. This is often the case in sport (and in turn cricket), for various reasons. For example, if attempting to model player performance when a player may have only played a handful of professional matches. This issue can be addressed using transfer learning or pseudo Monte Carlo sampling techniques.

1.3 Potential Solutions

A potential solution is to use an entirely different approach to those that are currently popular in sports analytics, namely Gaussian processes (GPs). GPs are particularly effective in handling small datasets and can model complex distributions. One of the main benefits of Gaussian processes is their ability to incorporate uncertainty into the model. This means that instead of providing a single prediction, Gaussian processes can provide a range of possible outcomes along with the associated probabilities.

Although neural networks and boosting algorithms also attempt to predict with uncertainty, they often rely on heuristic methods to estimate uncertainty, which may not be as accurate as a fully Bayesian approach. In contrast, Gaussian processes are inherently Bayesian, which means they can provide more accurate and reliable estimates of uncertainty.

1.3.1 Benefits of Gaussian Processes for Cricket Modeling

Cricket is a sport that involves many variables, including pitch conditions, weather, player skills, and tactics. These variables can all impact the outcome of a game, and as a result, it can be challenging to model the performance of a cricket team accurately. Additionally, there may be inherent noise in the training data, such as errors in data collection or variations in player performance. Furthermore, outputs can constantly overlap the same regions of input space. For these reasons, Gaussian processes are an underutilised model choice for cricket because they can handle noisy training data and can provide probabilistic predictions.

1.4 Project Plan

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 unforseen challenges, or sections that need it. See a Gantt chart for an overview of the plan in figure 1.1.
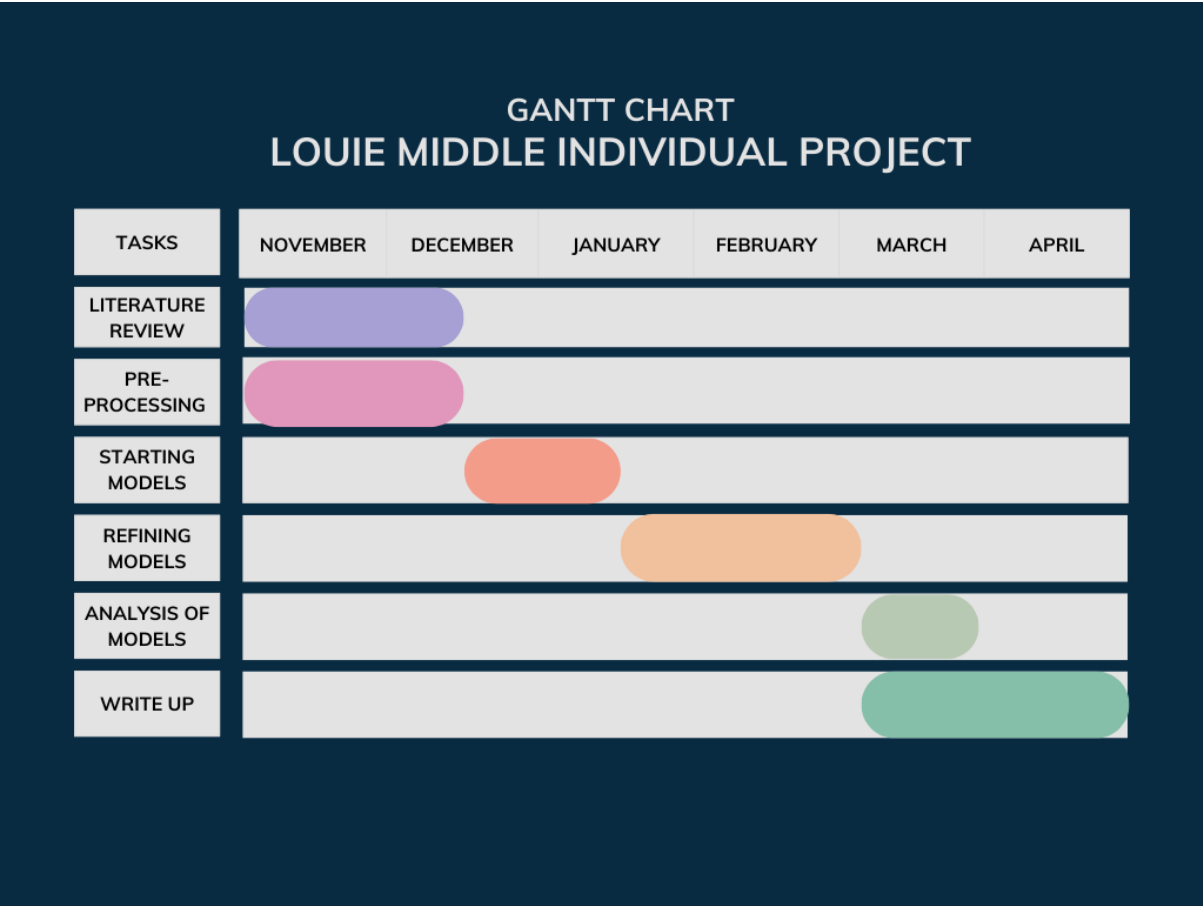
Figure 1.1: Gantt Chart For Project

1.5 Resources Required

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], at least 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 GPU cluster.

Chapter 2

Literature and Technology Survey

TODO: Talk about secretive nature of sports analytics (much of it is private).

2.1   Moneyball

The release of the book Moneyball [Lewis, 2004] and its subsequent film adaptation, 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 in 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. During this time 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. Attempting to replicate these results 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. Graham's 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, paired 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.

2.2   T20 Cricket

Whilst there are varying formats of cricket [1], the focus in this project is the T20 format (20 over cricket) that is played in the IPL. T20 cricket gives the first team 20 overs to score as many runs as possible, or until they have used all 10 of their wickets. The teams then swap roles and the opposing team attempt to "chase" the total runs plus one scored by the first team. If they manage to do so they win the match, but if all their batters get out before the target total, or use all 20 overs before reaching the target, they lose.

Longer formats of the game such as 40/50 over one day matches and four/five day matches have different tactics compared to T20 cricket. Due to T20s shorter nature, a bowling team's goal is generally to minimise the runs the batting team can score, in addition to the traditional goal of taking wickets. Typically, once a batter settles into their innings they attempt to score more boundaries to maximise the runs scored in the short playing period. Furthermore, some "power hitters" are selected for their specialism in scoring boundaries (see A.3.2) at the end of an innings, with no "settling in" period that is afforded to batters in longer formats or at the start of a T20 innings. Whilst this style of play is more risky as the chances of misplaying increase, it is generally more beneficial to score as many runs as possible in the short time T20 cricket allots. In general T20 batters protect their wicket less than those in other formats and play a riskier, high run scoring style of play. Irvine and Kennedy [2017] and the general consensus of the cricket community is that this high boundary scoring carries a greater importance than scoring singles (see A.3.2). Irvine and Kennedy [2017] found that the higher the number of singles scored was detrimental to a batting team's chances of success.

Due to the batters desire to score boundaries in T20, bowlers might adopt different strategies to reduce the batters ability to score boundaries. This is often done through bowling a wider variety of deliveries and aiming for dot balls (see A.3.2) or singles. Consequently, this should reduce the runs the batters score and increase the chance of taking a wicket due to the batter needing to play a wider variety of shots to score runs, which increases the chance of them making a mistake. In general, T20 bowling attacks often assess the ways to reduce the number of boundaries the opposing team can score. Irvine and Kennedy [2017] found that increased number of dot balls led to a higher chance of winning for the bowling team.

2.2.1   Indian Premier League

The Indian Premier League (IPL) is a professional cricket league based on the T20 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. This increases the desire to use modern machine learning techniques to improve results.

---

[1]Whilst knowledge of cricket is useful to understand this project, it is not necessary to appreciate the project and its applications. Despite this, I present a brief introduction to the game and its laws for those unfamiliar with the sport in chapter A.

2.3   Existing Machine Learning Research on Sport
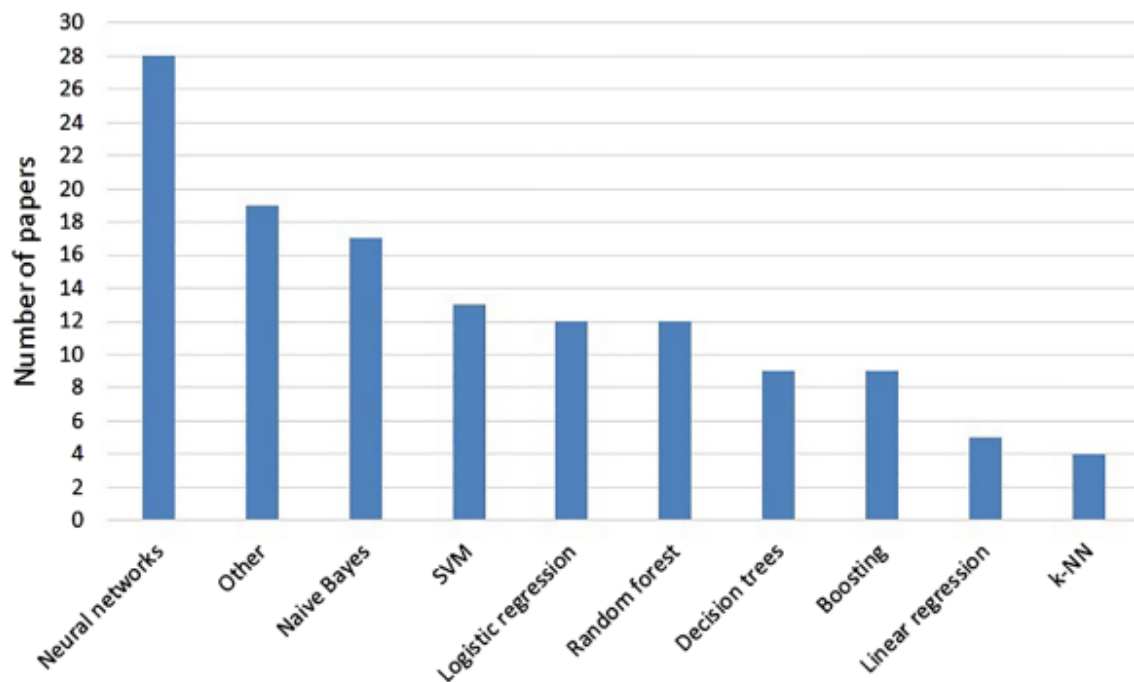
Figure 2.1: Total number of papers using a particular ML algorithm group [Horvat and Job, 2020]

Figure 2.1 from Horvat and Job [2020]'s literature review of machine learning in sport for score prediction show the methods that are currently used in the field as of May 2020. As shown in the figure, neural networks have a large research output. This is followed by other popular methods such as regression, support vector machines (SVMs), regression, decision trees, and gradient boosting, with k-nearest-neighbours (k-NN) being the least used. It is important to note that whilst a larger body of research has been carried out with certain methods, it does not necessarily mean they are the optimal method.

Horvat and Job [2020] also showed that including excessive 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 factors relating to team composition and tactics can change. The most accurate 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, the majority of the data was used for training and a small portion for testing. Some researchers used the same data set for training and testing, yielding unrealistically accurate results.

2.3.1   XGBoost and Football

TODO: Talk about [Blumberg, 2020].

TODO: Talk about stats bomb.

2.4 Existing Machine Learning Research on Cricket

TODO: Talk about CricketSavant, his XGBoost, and then subsequent hiring by CrickVis.

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.

Table 2.1: Research areas in cricket and their percentage of the total research output in cricket [Wickramasinghe, 2022]

Research Area	% of Studies
Game Outcome Prediction	35
Player's Performance Classification	16
Batting Style/Stroke Classification	9
Other	9
Umpire's Decision/Gestures	6
Score Prediction	5
Cricket Commentary/Media	4
Pitch Behaviour Predicition	4
Team Selection/Performance	3

Wickramasinghe [2022] review of machine learning in cricket from December 2022 classified the current areas of research in cricket. He found that over half of studies aim to predict the outcome of a game, or classify different players performances. The full table of results can be seen in table 2.4. From this it is clear to see that there is very little research in predicting batter bowler matchups or the outcome of a single delivery. Instead, most of the focus is on predicting game outcomes or overall player performance.
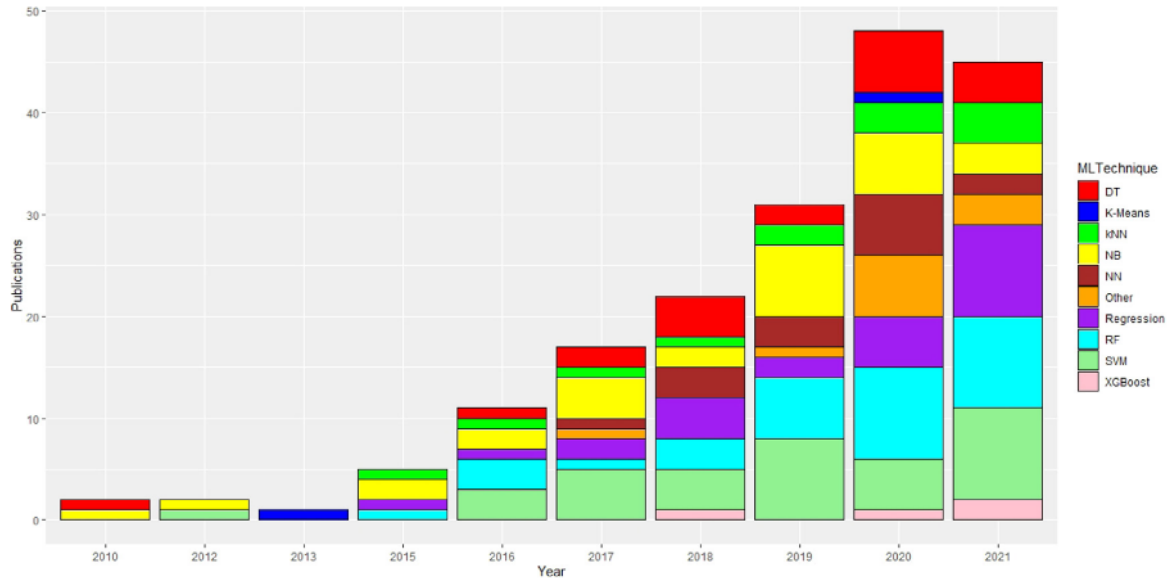
Figure 2.2: Total number of publications for each ML technique each year in cricket research [Wickramasinghe, 2022]

Wickramasinghe [2022] also evaluated the frequently used ML techniques in cricket, as shown in figure 2.2. There is a noticeable lack of deep learning, reinforcement learning or natural language processing techniques. Furthermore, better boosting algorithmgs such as XGBoost have only recently begun to be adopted, and Gaussian processes have not been used at all. Lastly, a clear uptick in the number of publications can be seen over the years, which is also corroborated in sport as whole in Horvat and Job [2020]. It once again highlights the growing interest in ML in sport.

2.4.1   Existing Machine Learning Research on the Indian Premier League

Saikia et al. [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%.

Otherwise, there is generally limited research on the IPL and it is an area ripe for research.

2.5   Research With Similar Goals

The Singlearity-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.

The Singlearity-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. SinglearityPA 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 Singlearity-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, Singlearity-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.

2.6 Gaussian Processes in Sport

TODO: Talk about Gaussian processes in sport

Chapter 3

Background

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

3.1 Gaussian Processes

[Yi, 2019] TODO: This has good explanations of stuff

[Griffiths, 2023] TODO: As does this

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 [Kaiser, 2017].

In addition, Gaussian processes are a supervised learning technique. It starts with a training data set $\mathcal{D}$ of n observations, $\mathcal{D} = (\mathbf{x}_i, y_i | i = 1, ..., n)$. $\mathbf{x}$ 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 Dxn 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

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 finite 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 $cov[x_i, x_j] = \Sigma_{ij}$.

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

$$
\begin{aligned}
m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\
k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))],
\end{aligned}
\tag{3.1}
$$

where we will write the Gaussian process as

$$
f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).
\tag{3.2}
$$

The covariance function, or 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 that 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 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 [Kaiser, 2017]. A kernel that only depends on the distance between two points is called *stationary*. Conversely, kernels that do depend on two points' positions 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). For a finite dimensional input space $\mathbb{R}^d$, the SE kernel is defined by

$$
cov(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 exp(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Lambda^{-1}(\mathbf{x} - \mathbf{x}')).
\tag{3.3}
$$

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. $\Lambda^{-1} = diag(l_1^2, ..., l_d^2)$ is a diagonal matrix of the squared length scales $l_i \in \mathbb{R}_{>0}$ in each axis. The variable $\sigma_f^2 \in \mathbb{R}_{>0}$ is the signal variance. The signal variance specifies the average distance of function values from the mean value. The vector of all hyperparameters in a model is called $\boldsymbol{\theta}$. Choices of such free hyperparameters will be discussed more in chapter 4 including their optimisation.
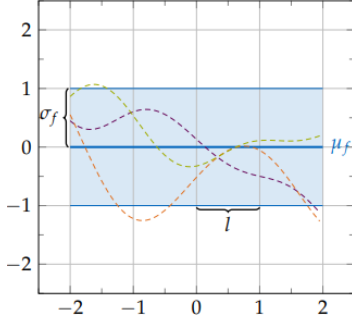
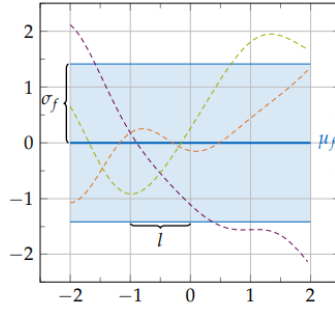Figure 3.1: SE kernel with $\sigma = 1$ and $l = 1$ [Kaiser, 2017]

Figure 3.2: SE kernel with $\sigma = \sqrt{2}$ and $l = 1$ [Kaiser, 2017]
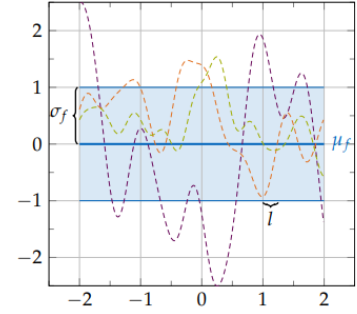
Figure 3.3: SE kernel with $\sigma = 1$ and $l = 0.25$ [Kaiser, 2017]

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. As can be seen in 3.3, reducing the length scale makes drawn sample functions from the kernel more "wiggly". Increasing the signal variance as shown in 3.2, increases the range of values the sample function will take.

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 *marginilisation of f*.

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

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right). \tag{3.4}$$

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

This results in the key predicitve equations for Gaussian process regression

$$\mathbf{f}_* | X, \mathbf{y}, X_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, cov(\mathbf{f}_*)) \tag{3.5}$$
$$\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_* | X, \mathbf{y}, X_*] = K(X, X_*)[K(X, X) + \sigma_n^2 I]^- 1 \mathbf{y} \tag{3.6}$$
$$cov(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^- 1 K(X, X_*). \tag{3.7}$$

Eq. 3.6 and eq. 3.7 represent the mean function and the covariance function of the Gaussian posterior process respectively.

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

Marginal Likelihood

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

$$\log p(\mathbf{y}|X) = -\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I)^-1\mathbf{y} - \frac{1}{2}log|K + \sigma_n^2 I| - \frac{n}{2}log 2\pi. \tag{3.8}$$

With this, we can create an algorithm for Gaussian process regression, which is given in alg. 1. The matrix inversion required by eq. 3.6 and 3.7 uses Cholesky factorisation, explained in section B.1.

Algorithm 1 Algorithm for Gaussian process regression [Rasmussen and Williams, 2006]

Require: X (inputs), $\mathbf{y}$ (targets), k (covariance function), σ_n^2 (noise level), X_* (test input)
  1: $L \leftarrow cholesky(K + \sigma_n^2 I)$
  2: $\boldsymbol{\alpha} \leftarrow L^T \setminus (L \setminus \mathbf{y})$ ▷ eq. 3.6
  3: $\bar{f}_* \leftarrow \mathbf{k}_*^T \boldsymbol{\alpha}$ ▷ eq. 3.6
  4: $\mathbf{v} \leftarrow L \setminus \mathbf{k}_*$ ▷ eq. 3.7
  5: $\mathbb{V}[f_*] \leftarrow k(X_*, X_*) - \mathbf{v}^T \mathbf{v}$ ▷ eq. 3.7
  6: $\log p(\mathbf{y}|X) \leftarrow -\frac{1}{2}\mathbf{y}^T\boldsymbol{\alpha} - \Sigma_i \log L_{ii} - \frac{n}{2}\log 2\pi$ ▷ eq. 3.8
  7: **return**: $\bar{f}_*$ (mean), $\mathbb{V}[f_*]$ (variance), $\log p(\mathbf{y}|X)$ (log marginal likelihood)

Loss Functions

For practical applications, there must be a decision on how to act - a point prediction is necessary. To achieve this, we need a loss function $\mathcal{L}(y_{true}, y_{guess})$ which specifies the loss incurred by guessing the value y_{guess} when the true value is y_{true}. The goal is to then make the point prediction y_{guess} that incurs the smallest loss. This can be done by minimising the expected loss or risk by averaging w.r.t. our model what the truth might be. Thus the best guess is

$$y_{optimal}|\mathbf{x}_* = \underset{y_{guess}}{\operatorname{argmin}} \widetilde{R}_{\mathcal{L}}(y_{guess}|\mathbf{x}_*) \tag{3.9}$$

where $\widetilde{R}_{\mathcal{L}}$ represents the risk and $\widetilde{R}_{\mathcal{L}}(y_{guess}|\mathbf{x}_*) = \int \mathcal{L}(y_*, y_{guess}) p(y_*|x_*, \mathcal{D}) dy_*$.

Choosing Hyperparameters

Up until now, the hyperparameters $\boldsymbol{\theta}$ have been assumed constant. If this were the case, GPs would have no training stage. However, knowing the correct hyperparameters is not clear a priori, and such a method of determing the hyperparameters must be used.

The correct way to model uncertainty about the hyperparameters is to give them a prior probability $p(\boldsymbol{\theta})$. The prior chosen should be broad to reflect the vagueness of the parameters before training. To derive the dependent distributions marginalise the prior to get

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

A new distribution can be obtained by combining the prior with the likelihood of the training data observed using Baye's theorem:

$$
\begin{aligned}
p(\boldsymbol{\theta}|X, y) &= \frac{p(y|X, \theta)p(\theta)}{p(y|X)} \\
&= \frac{p(y|X, \theta)p(\theta)}{\int p(y|X, \theta)p(\theta)d\theta}
\end{aligned}
\tag{3.11}
$$

The integration required in the denominator of eq. 3.11 is very hard in practise as y is a complicated function of $\boldsymbol{\theta}$. Instead, $p(\boldsymbol{\theta}|X, y)$ is maximised to provide an estimate. The solution of $p(y|X, \theta)$ you might recognise as the log marginal likelihood in eq. 3.8. For practical reasons minimising the negative of the log marginal likelihood is easier than maximising it directly. Therefore, the estimation of hyperparameters is the solution of the following optimisation problem:

$$\boldsymbol{\theta}^* \in \operatorname*{argmin}_{\boldsymbol{\theta}} - p(y|X, \theta). \tag{3.12}$$

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 or multi-class likelihood must be used for approximate inference.

In such cases the latent function f plays the role of a nuisance function. This is that values of f are not observed directly and there is no interest in the values of f, but rather in π where $\pi(\mathbf{x})$ is TODO. The purpose of f is solely to allow convenient formulation of the model and the goal is to integrate out f.

TODO: Explain variational GPs

3.1.5 Sparse Variational Gaussian Processes

A major drawback of Gaussian processes is their $O(N^3)$ complexity when computing $[K + \sigma_n^2 I]^{-1}$ from 1, but it 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 $O(N^2)$. An approach to reduce this complexity is *Sparse Variational Gaussian Processes* (SVGP) [Hensman et al., 2014]. Spare approximations of GPs are a method of approximating a GP using a $M < N$ set of *inducing points* $\mathbf{Z}$ and *inducing variables* $\mathbf{u}$ that can represent the entire dataset, rather than X itself. This approach

suits datasets with high levels of redundancy, but also introduces the problem of choosing the subset.

Using inducing points, we arrive at the Nyström approximation

$$K_{NN} \approx K_{NM} K_{MM}^{-1} K_{MN}, \tag{3.13}$$

where $K_{NM} = K(\mathbf{X}, \mathbf{Z})$ and $K_{MM} = K(\mathbf{Z}, \mathbf{Z})$. The SVGP complexity now costs $O(NM^2)$ instead of $O(N^3)$.

When the inducing points are the real data points, such that $\mathbf{Z} = \mathbf{X}$ then $K_{NM} = K_{MM} = K_{NN}$ and the bound on the marginal likelihood becomes tight. Therefore, the approximate marginal likelihood equals the real marginal likelihood.

Choosing the Inducing Points

When choosing your inducing points you want to use a method that will select M inducing points, that can represent the N real points. This can be done in many ways; an obvious example is to select inducing points linearly across the input range. However if your input data is not linearly spaced, then this can misrepresent your data. A better method used in this project and used in Hensman et al. [2014] is k-means clustering, which can appropriately find all major regions of the input space so long as there are enough inducing points.

TODO: Maybe visualise this?

3.2   Data Association
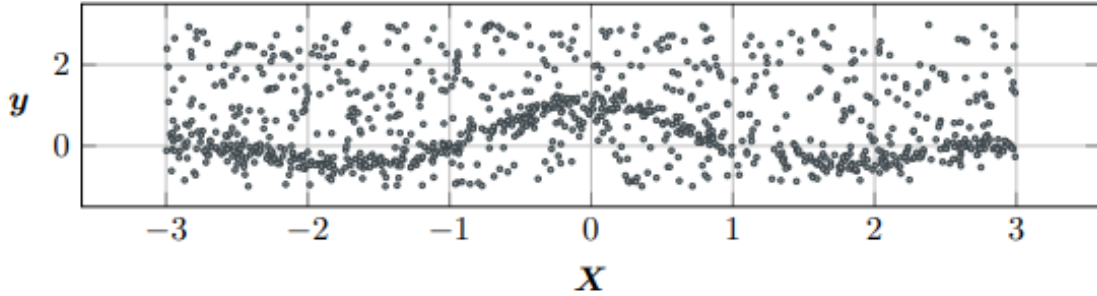
Figure 3.4: A data association problem consisting of two generating processes, one of which is a signal to recover and one is an uncorrelated noise process [Kaiser et al., 2018]

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 asymettric noise. Standard machine learning approaches can pollute any models, where the model starts to explain the noise instead of the underlying signal.

3.2.1   Data Association and Cricket

The cricket dataset could be deemed a data association problem, as perhaps we can say that each outcome (runs scored) of a cricket delivery is generated by a different process. Each of these processes are noisy, as cricket has stochastic outcomes.

Furthermore, the relationship between a cricket delivery's features and its outcome is not a simple relationship. For example, lets take a high level look at a seam bowlers length of pitch in IPL cricket (the "pitchY" feature in the dataset). A good short ball (also known as a bouncer) from a quick bowler is one that reaches the batsmen at shoulder or above height. Any delivery pitching slightly fuller will be at the batsmens chest, which is considered a poor ball, as it is far easier for the batsmen to get over with their bat and hit for boundaries. Equally, a slightly fuller pitch again puts the ball back in a good length, as it is around the top of the stumps. Slighly fuller however, is once again deemed poor, as it is at a full length for the batsmen to step forward and drive or slog for runs. Yet, once again, slighly fuller is a good length. This is as a yorker, a ball right at a batsmens feet, which is considered a very difficult ball to play against. Therefore it is clear the relationship between the pitch length and the outcome of a delivery is not linear and not simple. Similar examples can be made with the width of a ball, its swing, its spin, and many other features.

It is clear that the game of cricket has many nuances and challenges for modelling with machine learning techniques. Data association techniques however, provide a framework to attempt to seperate some of the differences in how runs are scored in cricket, and model the overlap in the varying outcomes.

3.3   XGBoost

XGBoost is a scalable machine learning gradient boosting library. It is an extremely popular, versatile and state-of-the-art gradient boosting technique. For example, among the 29 challenge winning solutions published at Kaggle's blog during 2015 17 solutions used XGBoost. Among these solutions, eight solely used XGBoost to train the model, while most others combined XGBoost with neural nets in ensembles. These results demonstrate that XGBoost gives state-of-the-art results on a wide range of problems. Examples of the problems in the winning solutions include: store sales prediction; high energy physics event classification; web text classification; customer behavior prediction; motion detection; ad click through rate prediction; malware classification; product categorization; hazard risk prediction; massive online course dropout rate prediction [Chen and Guestrin, 2016].

The versatility and performance of XGBoost make it a popular choice of boosting algorithm, yet it hasn't been widely adopted in cricket research (see section 2.4). For this reason, I wanted to experiment with XGBoost in this project, and use it as a baseline performance compared to solutions with GPs.

3.4   Hawk-Eye and Ball Tracking

The data-set used in this project contains tracking data from Hawk-Eye systems. Hawk-Eye Innovations is a sports technology company, that provides many products, including computer version services. In particular there is a focus on ball and player tracking in sports, including cricket. In this project the emphasis is placed on the ball tracking. Hawk-Eye have a number

of cameras at IPL grounds, that the video from is then triangulated and combined to create a three-dimensional representation of the ball's trajectory. It has been adopted as an ajudication tool by many governing bodies in cricket and is generally accepted to be accurate.

Hawk-Eye tracking data is accumulated and stored. It can then be used for various data science and machine learning purposes. Further explanation of the dataset and features used in this project are explained in section 4.2.

Chapter 4

Method

4.1 Objectives

Following on from the appropriate literature review and background, the primary objectives of this project can be listed as follows:

1. From 2.2 it is clear that it is key for bowling attacks to reduce the number of boundaries scored and increase the number of dot balls. Therefore, a key objective of this project is to accurately predict which delivery would be most optimal to stop a boundary and cause a dot ball.

2. From sections 2.3 and 2.4, it is clear that there is a lack of current research in delivery outcome prediction and use of a wider breadth of ML techniques. Another objective of this project, is to attempt to explore new prediciton types in cricket, as well as use under-utilised ML techniques, notably:

    (a) Mixtures of Gaussian processes.

    (b) XGBoost

4.2 Dataset

The dataset used is the mens IPL cricket dataset. The dataset is a CSV where each row is a different cricket delivery. Each row contains identifying information of the delivery, such as the match ID, delivery number, the batter, the bowler, and additional Hawk-Eye data. For this project, personal information on whose batting or bowling will not be disclosed. Hawk-Eye data includes the bowling style, ball speed, the runs scored (including how many were from the batter or extras), where the ball pitches, where the ball passed the stumps and more.
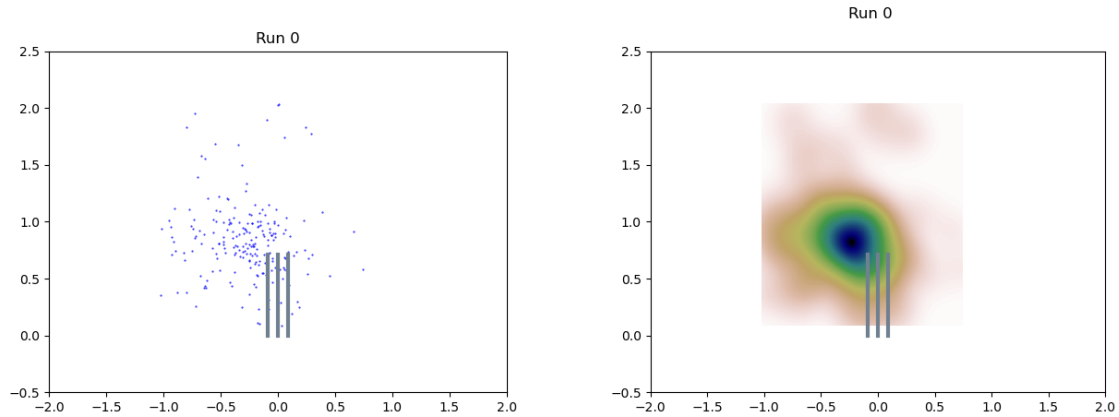
Figure 4.1: The "StumpsX" and "StumpsY" features of the dataset. This is all the right arm seam deliveries that went for 0 "batterRuns" for one batter.
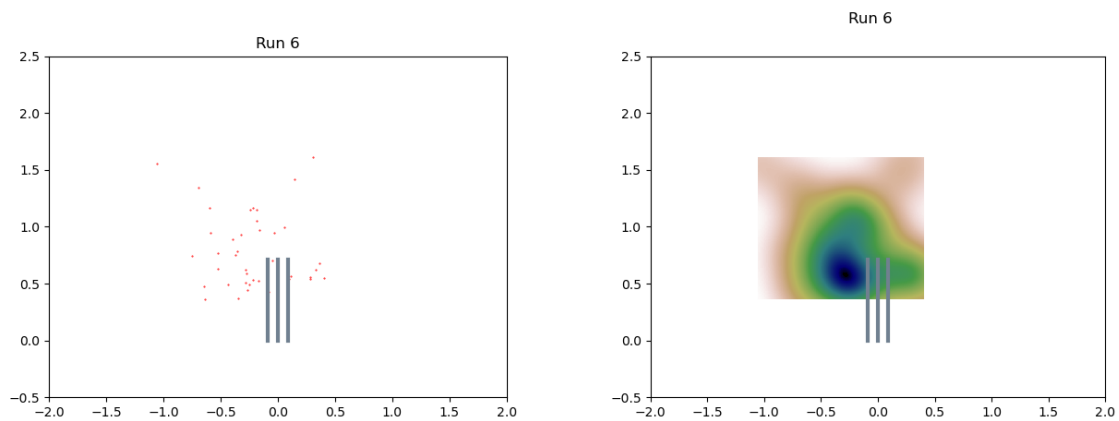
Figure 4.2: The "StumpsX" and "StumpsY" features of the dataset. This is all the right arm seam deliveries that went for 6 "batterRuns" for one batter.

One noticeable thing is how much the data overlaps for different outcomes. Whilst their most likely is relationships and functions that can describe good and bad balls, it is clear that cricket is very noisy.

4.3   Mixture of Gaussian Processes

4.3.1   Data Association with Gaussian Processes

The data association with Gaussian processes model introduced by Kaiser et al. [2018] assumes there exists K independent functions. The assignment of the nth data point to a function is specified by the indicator vector $\mathbf{a}_n \in {0,1}^K$ which has exactly one non-zero entry. The assignment probabilities of outputs is calculated with a softmax function and multinomial distribution.

4.3.2   Modulated Scalable Gaussian Processes

Liu et al. [2020] proposed a similar method with a tighter evidence lower bound (ELBO). Kaiser et al. [2018] keeps the latent variables $\mathbf{W}$ by approximating $\log p(\mathbf{y}, \mathbf{W})$ rather than the interested $\log p(y)$. Liu et al. [2020] directly approximates $\log p(y)$ by marginalising all the latent variables out.

The scalable mixture of Gausssian processes model developed by Liu et al. [2020] makes use of the reparametisation trick [Maddison et al., 2017]. TODO: Fill this out.

4.3.3   Implementation

The implementation to mix gaussian processes was an iteration of the SMGP (scalable modulated Gaussian processes) model developed by Liu et al. [2020]. The original model was developed in TensorFlow version one and code copied over from GPFlow version one [Matthews et al., 2017]. This was then updated to use the latest GPFlow API version two and TensorFlow version two. Following this the code could now be run in eager execution, enabling debugging. The code was then extended to enable different likelihoods and multiple inputs.

TODO: Describe implementation and approach to SMGP model

4.4   XGBoost

The XGBoost [Chen and Guestrin, 2016] model implemented was a modification of the XGFootball model developed by Blumberg [2020]. In the code published by Blumberg, there is an implementation of various boosting and decision tree models. Rather than train every classifier used by Blumberg, only the XGBoost model was trained on the cricket dataset.

TODO: Describe implementation and approach to XGBoost model

Chapter 5

Results

5.1 Mixture of Gaussian Processes

TODO: Write up results once assignment function of SMGP smooth

5.2 XGBoost

TODO: Write up results of Jos Buttler demo from DoP.

Chapter 6

Discussion

TODO: Discuss results.

Chapter 7

Conclusion

TODO: Conclude on results and findings. Explain future work (e.g. taking into account that we know the assignments). Furthermore more inputs and use of categorical inputs etc.

7.1   Future Work

Bibliography

Tomislav Horvat and Josip Job. The use of machine learning in sport outcome prediction: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(5):e1380, 2020. URL `https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1380?casa_token=aXIDtR-gOHQAAAAA%3A-YGsS9bhdiOkYkurvCmBCdZpbAx-MV4VF806rtp_h1CXJlCCOpg4hFFknpmc9dAb3wBAVu7NZ_FOBQ`.

Indika Wickramasinghe. Applications of machine learning in cricket: A systematic review. *Machine Learning with Applications*, 10:100435, 2022. URL `https://www.sciencedirect.com/science/article/pii/S2666827022001104`.

Markus Kaiser. Incorporating uncertainty into reinforcement learning through gaussian processes. Master's thesis, 2017. URL `https://github.com/mrksr/masters-thesis/blob/master/thesis/thesis.pdf`.

Markus Kaiser, Clemens Otte, Thomas Runkler, and Carl Henrik Ek. Data association with gaussian processes, 2018. URL `https://arxiv.org/abs/1810.07158`.

Wikipedia, 2023a. URL `https://en.wikipedia.org/wiki/Cricket#`. Last accessed 18 April 2023.

Wikipedia, 2023b. URL `https://en.wikipedia.org/wiki/Leg_before_wicket#`. Last accessed 18 April 2023.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Process for Machine Learning*, volume 2. MIT Press Cambridge, MA, 2006. URL `http://gaussianprocess.org/gpml/`.

Michael Lewis. *Moneyball: The art of winning an unfair game*. WW Norton & Company, 2004.

Fox Sports. The tampa bays are the undisputed kings of moneyball, 2019. URL `https://www.foxsports.com/stories/mlb/the-tampa-bay-rays-are-the-undisputed-kings-of-moneyball`.

Liverpool FC News. Who is ian graham? liverpool suffer second transfer blow after julian ward departure, 2022. URL `https://www.liverpool.com/liverpool-fc-news/features/ian-graham-liverpool-transfer-ward-25596815`.

Liverpool Echo. The unknown liverpool backroom analyst who convinced jurgen klopp to commission transfers of salah and keita, 2019. URL `https://www.liverpoolecho.co.uk/sport/football/transfer-news/unknown-liverpool-backroom-analyst-who-16320524`.

Scott Irvine and Rodney Kennedy. Analysis of performance indicators that most significantly

affect international twenty20 cricket. *International Journal of Performance Analysis in Sport*, 17(3):350–359, 2017. URL `https://doi.org/10.1080/24748668.2017.1343989`.

ESPNcricinfo. How can the ipl become a global sports giant?, 2018. URL `https://www.espncricinfo.com/story/tim-wigmore-how-can-the-ipl-become-a-global-sports-giant-1150459`.

Rob Blumberg. A fully transparent expected-goals model, 2020. URL `https://robblumberg.github.io/exp-goals/`.

Stylianos Kampakis and William Thomas. Using machine learning to predict the outcome of english county twenty over cricket matches, 2015. URL `https://arxiv.org/ftp/arxiv/papers/1511/1511.05837.pdf`.

Hemanta Saikia, Dibyojyoti Bhattacharjee, and Hermanus H Lemmer. Predicting the performance of bowlers in ipl: An application of artificial neural network. *International Journal of Performance Analysis in Sport*, 12(1):75–90, April 2012.

Joshua Silver and Tate Huffman. Baseball predictions and strategies using explainable ai. In *The 15th Annual MIT Sloan Sports Analytics Conference*, 2021. URL `https://singlearity.com/static/assets/sloan-singlearity.pdf`.

Wei Yi. Understanding gaussian process, the socratic way, 2019. URL `https://towardsdatascience.com/understanding-gaussian-process-the-socratic-way-ba02369d804`.

Ryan-Rhys Griffiths. Applications of gaussian processes at extreme lengthscales: From molecules to black holes, 2023. URL `https://arxiv.org/pdf/2303.14291.pdf`.

James Hensman, Alex Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification, 2014. URL `https://arxiv.org/pdf/1411.2005.pdf`.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Haitao Liu, Yew-Soon Ong, Xiaomo Jiang, and Xiaofang Wang. Modulating scalable gaussian processes for expressive statistical learning, 2020. URL `https://arxiv.org/abs/2008.12922`.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables, 2017. URL `https://arxiv.org/pdf/1611.00712.pdf`.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017. URL `http://jmlr.org/papers/v18/16-537.html`.

Appendix A

Cricket Background

Whilst knowledge of cricket is useful to understand this project, I do not deem it necessary to appreciate the project and its applications. Any key ideas around cricket, T20 and IPL beyond the laws of the game are introduced in the main report. Despite this, I present a brief introduction to the game and its laws for those unfamiliar with the sport. Certain aspects of cricket have been introduced in limited detail, such as fielding and umpiring as they are unnecessary to understand in detail for this project.

A.1   A Brief Overview of Cricket

Cricket is a bat and ball game played between two teams of eleven players. It is played on a field, the centre of which is a pitch with wickets at each end comprising two bails balanced on three stumps. At the edge of the field is a boundary rope, see section A.3.2. The batting side scores runs by hitting the ball bowled at one of the wickets and then running between the wickets, or scoring a boundary. The fielding side tries to prevent this, by getting the ball to either wicket and dismissing the batters so they are "out", see section A.3.4. Which side bats/fields first is determined before the match starts through a coin toss between both team captains. When ten batters have been dismissed, the innings ends and the teams swap sides (the fielding side bat, and the batting side field). The game is referreed by two umpires. In the case of international or professional cricket, there is typically an additional 3rd umpire off the field for video or technology assisted reviews.

A.2   Additional T20 Cricket Details

The following section includes any additional details on T20 cricket not included in section 2.2.

A.2.1   Super Overs

On the rare case a T20 game finishes with both sides scoring the same number of runs, a tie breaker "super over" is played. In a super over, each team nominates three batsmen and one bowler to play a one-over-per-side "mini-match". If the Super Over also ends up in a tie, it is repeated until the tie is broken. An *over* is a set of 6 fair deliveries (so not wides and no balls, see section A.3.3) for the batting team to score.
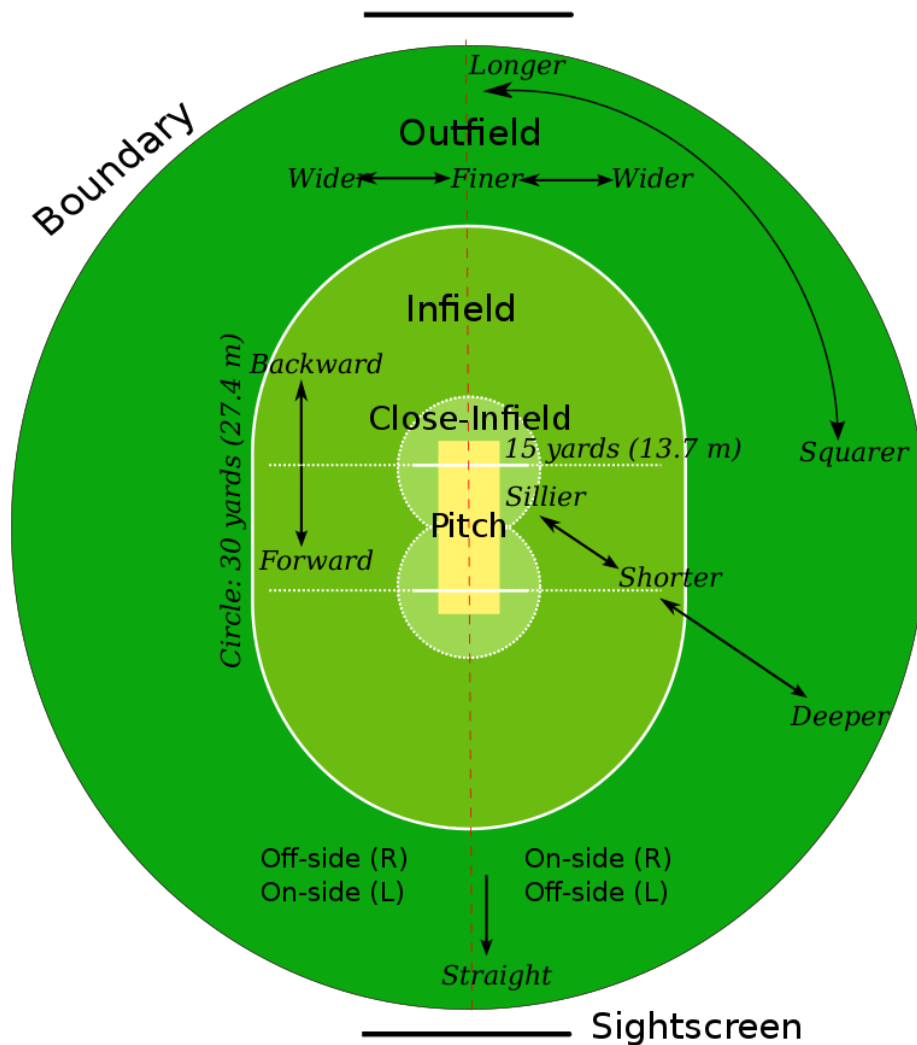
A.3 Laws and Gameplay

A.3.1 Playing Field

Figure A.1: A typical cricket field [Wikipedia, 2023a]

A cricket field as mentioned, comprises a central pitch with wickets at both ends of the pitch. A wicket is three stumps, with two bails balanced on top. See figure A.1 for a visual representation. The boundary surrounds the edge of the field and typically the field is oval in shape.
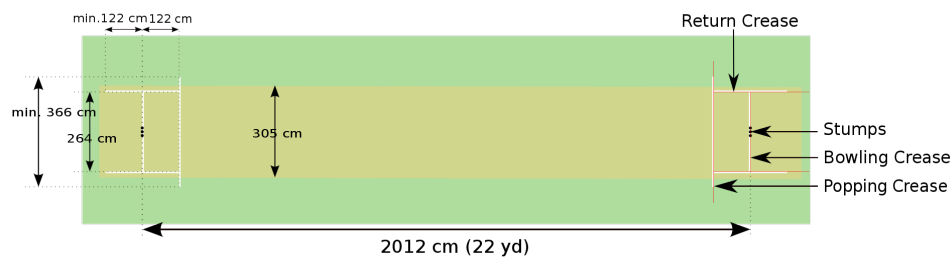
Figure A.2: A typical cricket pitch [Wikipedia, 2023a]

Figure A.2 shows a cricket pitch including the location of the wickets and the creases. As illustrated, the pitch is marked at each end with four white painted lines: a bowling crease, a popping crease and two return creases.

A.3.2   Runs

Runs are scored through either boundaries, or batsmen running between the creases. For the latter, the batter on strike attempts to score runs by hitting the ball whilst simultaneously not getting out. To score runs, after the batsmen strikes the ball there needs to be enough time for the batsmen at both ends to run to the other end of the wicket. If the fielding team get the ball to the wickets before the batsmen they could be *run out*, see section A.3.4. To register a run, both runners must touch the ground behind the popping crease with either their bats or their bodies (the batters carry their bats as they run). Each completed run increments the score of both the team and the striker. Infinitely many runs can be scored off one delivery this way, with the batsmen running back and forth between the wickets. Typically the maxmimum runs scored this way would be three, but it is possible to score more if there are misfields.

TODO: Explain definition of dot ball and single.

Boundaries

A boundary is scored when the ball is hit by the batsmen all the way to the boundary rope. Four runs are scored if the ball touches the ground at all prior to crossing the rope. If the ball is hit all the way over the boundary rope before touching the ground this is six runs. In T20 cricket the batsmen will typically aim to score boundaries, as they have the maximum return of runs.

Extras

Additional runs can be gained by the batting team as extras due to errors made by the fielding team. This is achieved in four ways: no-balls, wides, byes, and leg byes. A no ball occurs if the bowler bowls an illegal delivery, most commonly by overstepping the crease, but can also be called by an umpire for dangerous bowling. In T20 cricket and IPL, no balls award the batting team a *free hit*, a delivery in which they can only be dismissed via run out, hit the ball twice and obstructing the field (see section A.3.4). A wide has occured if the bowler bowls so that the ball is out of reach of the batter. Both a wide and a no ball have to be re-bowled and award the batting team one run. Byes are any runs the batters achieve without hitting

the ball, typically when the ball has been missed by the wicket keeper. Leg byes are any runs the batters achieve when the ball hits their body, but not their bat.

A.3.3   Overs

An over is 6 fair deliveries (not including wides and no balls etc.). A bowler can not bowl 2 overs in a row and at the end of an over the bowling end on the pitch changes. Despite this, typically bowlers will bowl in *spells*, where they will bowl alternate overs from the same end. In T20 cricket a bowler can bowl a maximum of 4 overs, meaning a minimum of 5 bowlers must be used. Typically a bowler might bowl two 2 over spells, however sometimes spinners in the middle of an innings might bowl all of their 4 overs in one spell. Unlike bowlers, the batsmen do not change ends at the end of each over.

A.3.4   Dismissals

There are nine ways a batter can be dismissed in cricket. The most common include being bowled, caught, leg before wicket (lbw), run out and stumped. Less common dismissals include hit wicket, hit the ball twice, obstructing the field, and timed out. For the purpose of this chapter I will only explain the most common dismissals.

Figure A.3: A clear example of leg before wicket (lbw) [Wikipedia, 2023b]

Bowled

Being bowled is when the batter misses the delivery and the ball hits the stumps and takes the bails off the wickets. If the bails do not come off the wickets, the batter is not out, though this is very rare.

Caught

Being caught is when the batter strikes the ball in the air and a fielder catches this before the ball touches the ground. If the ball touches the ground at any stage (even in the fielders hands) it is not out.

Leg Before Wicket (lbw)

Leg before wicket is a more complicated dimissal type. Following an appeal by the fielding side, the umpire may rule a batter out lbw if the ball would have struck the wicket but was instead intercepted by any part of the batter's body (except the hand holding the bat). The umpire's decision will depend on a number of criteria, including where the ball pitched, whether the ball hit in line with the wickets, the ball's expected future trajectory after hitting the batsman, and whether the batter was attempting to hit the ball. Without explaining the finer details of lbw, a clear example of lbw is illustrated in figure A.3. Further information on lbw can be found on Wikipedia [2023b].

Run Out

A run out usually occurs when the batters are attempting to run between the wickets. The fielding team must successfully get the ball to one wicket and take the bails off with the ball (or their hands with the ball in them) before a batter has crossed the crease line near the wicket. The incomplete run the batters were attempting does not count.

Stumped

Lastly, being stumped involves the wicket-keeper catching the ball after the delivery and taking the bails of the stumps with the ball (or their hands with the ball in them) while the batsman is out of his ground (the batsman leaves his ground when he has moved down the pitch beyond the popping crease, usually in an attempt to hit the ball). A batter can only be stumped off a fair delivery (not a no ball or a wide). It is a special case of being run out and can only be performed by the fielding wicket keeper.

A.3.5 Basic Gameplay Example

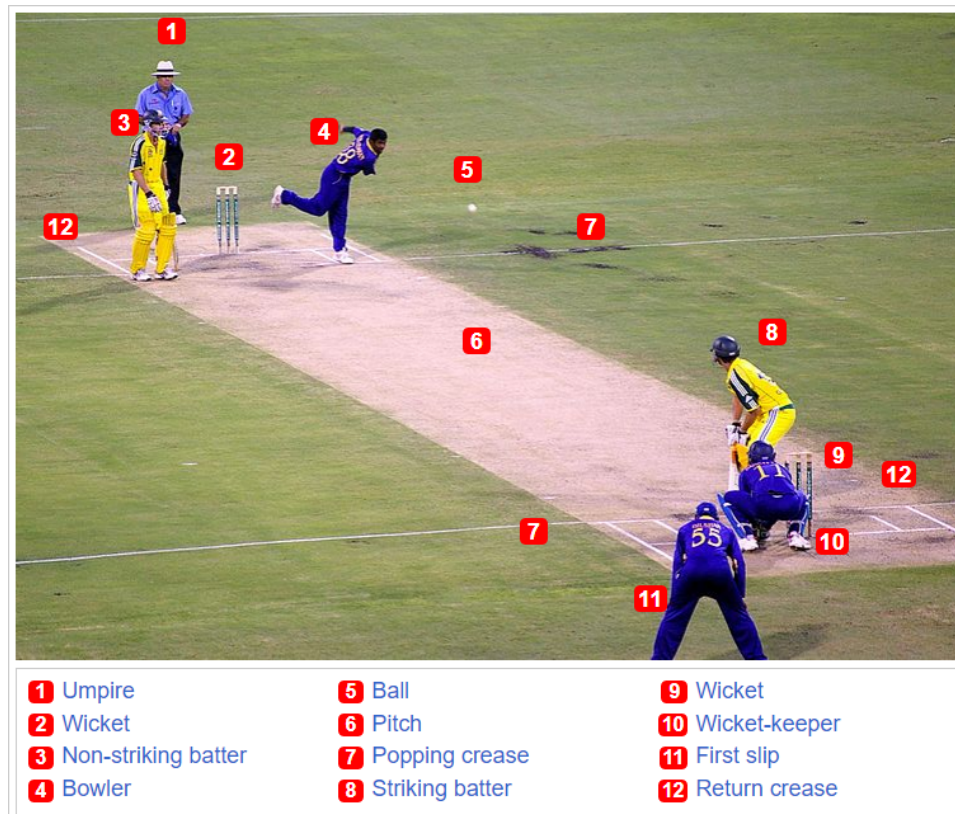|					
1 Umpire		**5** Ball		**9** Wicket	
2 Wicket		**6** Pitch		**10** Wicket-keeper	
3 Non-striking batter		**7** Popping crease		**11** First slip	
4 Bowler		**8** Striking batter		**12** Return crease	

Figure A.4: An example of a ball being bowled and the components in play [Wikipedia, 2023a]

Figure A.4 shows a cricket delivery in play. The two batters (3 and 8; wearing yellow) have taken position at each end of the pitch (6). Three members of the fielding team (4, 10 and 11; wearing dark blue) are in shot. The bowler (4) is bowling the ball (5) from his end of the pitch to the batter (8) at the other end who is called the "striker". The other batter (3) at the bowling end is called the "non-striker". The wicket-keeper (10) is positioned behind the striker's wicket (9).

The bowler (4) intends to dimiss the batsmen, or to prevent the striker (8) from scoring runs. By using his bat, the striker (8) intends to defend his wicket and hit the ball away from the pitch in order to score runs.

A.3.6 Player Roles

Typically players are selected to perform a specialised role.

Bowlers

As mentioned a minimum of five bowlers are required to bowl in a T20 match. Therefore, team selectors will typically choose 5-6 bowlers for the team for a given match. A bowler is someone specialised for bowling in one of two main ways: Seam (pace) bowling, or spin bowling. Seam bowlers typically use techniques such as *swing* or *seam* paired with a higher bowling speed (generally anywhere from 70 - 95 mph) to try and get wickets or reduce the

runs scored. However, in T20 cricket, seam bowlers will typically use a wider variety of delivery types, including slower balls, yorkers, bouncers, slower ball bouncers, wide yorkers, cross seam and more, to make prediciting their bowling more difficult and reduce the runs scored. Spin bowlers come in two main varieties: off spinners or leg spinners, and this essentially means what way they spin the ball. To spin the ball means to get the ball the ball to move dramatically in one direction after its bounce, by spinning it on release from the bowlers hand. Due to trying to spin the ball, spinners will typically bowl at a slower speed than pace bowlers (generally 45 - 65 mph). Like pace bowlers, a spin bowler in T20 cricket will use a variety of delivery types to try and reduce the runs scored from the opposing team.

Batters

The other main role is that of a batter. A batters job is to score runs for their team. In T20 cricket batters are generally specialised to score boundaries due to the short format, rather than be particularly good at protecting their wicket and staying in bat for many hours.

Wicket Keepers

The wicket keeper is a specialist fielder who stands behind the stumps to field the ball after it has been bowled. In the modern game, they are expected to also be reasonably good batsmen regardless of their wicket keeping skill.

All Rounders

All rounders are a special player who are good at both batting and bowling. All rounders are valued players, as they can provide the role of both a batter and bowler whilst only taking up one space on the teamsheet.

Captain

The last specialist role I will cover is that of the captain. The captain performs their captain duties as a batter, bowler, wicket keeper, or all rounder, while taking on the additional captains duties. The captain decides who will bowl each over and where each fielder will be positioned. While decisions are often collaborative, the captain has the final say. Captains in cricket typically shoulder more responsibility on the outcome of a cricket match than captains in other sports, given their level of responsibilty.

Appendix B

Mathematical Background

B.1   Cholesky Factorisation

B.2   Kullback–Leibler Divergence

Appendix C

Design Diagrams

Appendix D

Raw Results Output

Appendix E

Code

E.1 File: yourCodeFile.java

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

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

}
```