Applying Gaussian Processes in Cricket

Louie Middle

Bachelor of Science in Computer Science
The University of Bath
2023

Applying Gaussian Processes in Cricket

Submitted by: Louie Middle

Copyright

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. 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 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 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 T20 Cricket

Whilst knowledge of cricket is useful to understand this project, I do not deem it 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 for those wanting more information.

There are varying formats of cricket and the focus in this project is the T20 format (20 over cricket) that is played in the IPL. T20 cricket gives the first side 20 overs to score as many runs as they can, 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 they get all out before the target total, or use all 20 overs before reaching the target, they lose.

Whilst in longer formats of the game such as 40/50 over one day matches and four/five day matches the goal of the bowling side is to generally to take wickets, T20 cricket is different due to its shorter nature. In T20 cricket a bowling sides goal is generally to reduce the runs the batting team can score. Early on in a batters innings the best way to achieve this is to bowl good wicket taking deliveries, meaning balls around the stumps or with the potential to edge to the wicket keeper or slip fielders. This generally means straighter and arguably more predictable bowling. However, often once a batter settles into their innings they start to score many more boundaries off these types of deliveries. Whilst for the batter this is a more dangerous way to play as the chances of getting caught or misplaying go up, it is generally more beneficial to play like this and score as many runs as possible in the short time T20 cricket allots. This is because even if one or two batters get out this way, the chances of the entire team getting out this way is low. With the goal of scoring as many runs as possible, in T20 cricket batters therefore in general protect their wicket less and play a riskier, explosive, high run scoring style of play.

Due to this, bowlers might adopt different strategies to constrict the batters ability to score runs after they settle into their innings, often through bowling a wider variety of deliveries. Consequently, this should reduce the runs the batters score and increase their winnings chances, and might even take a wicket due to the batter needing to play riskier shots to score runs. Therefore in general in T20 cricket, bowling sides often look at ways to reduce the number of boundaries the opposing team can score. A key objective of this project is to accurately predict, what type of delivery would be best to restrict the batting team from scoring a boundary.

1.2.1 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 to see, thus increasing the desire to use modern machine learning techniques to improve results.

1.3 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 road bumps, or sections that need it. See Gantt chart for overview of plan in figure 1.1.

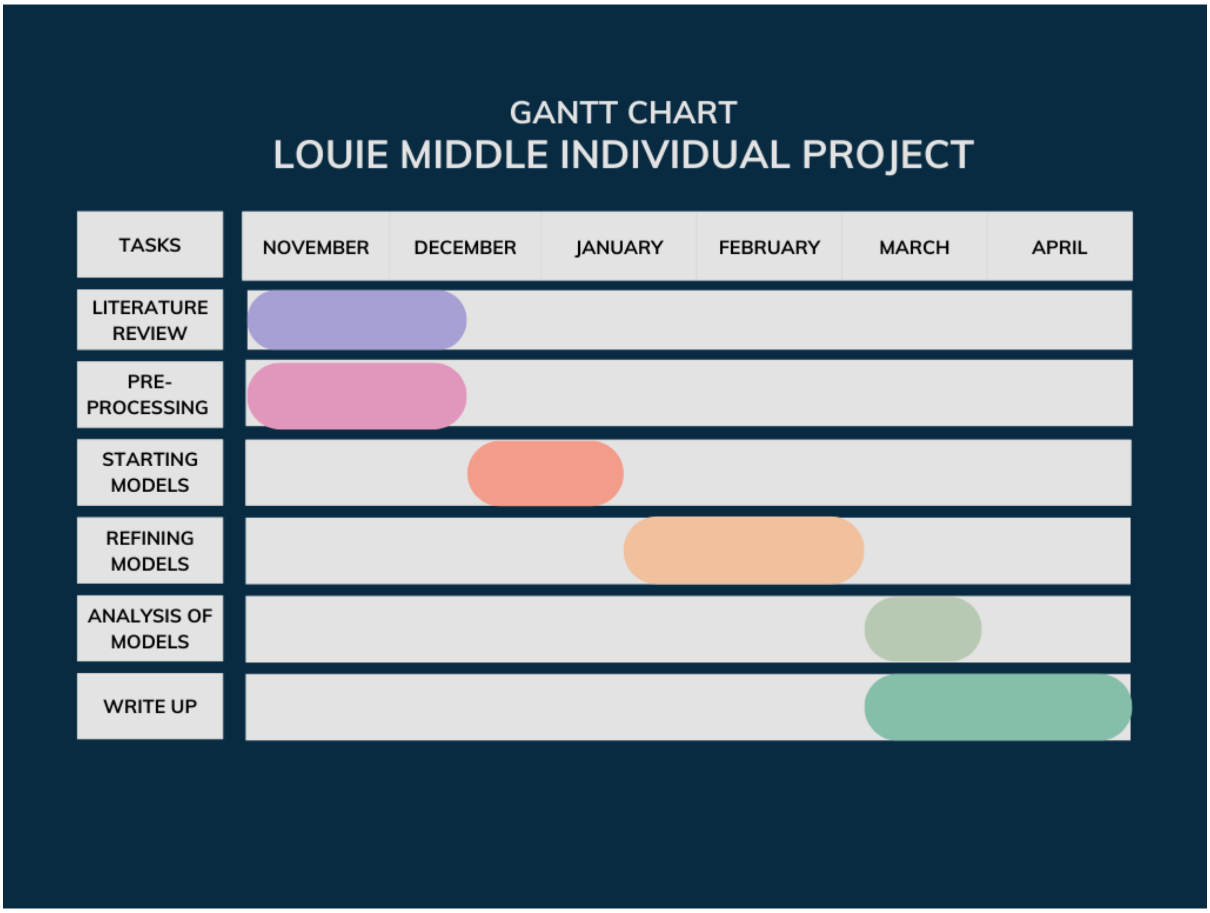

Figure 1.1: Gantt Chart For Project

1.4 Resources Required

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

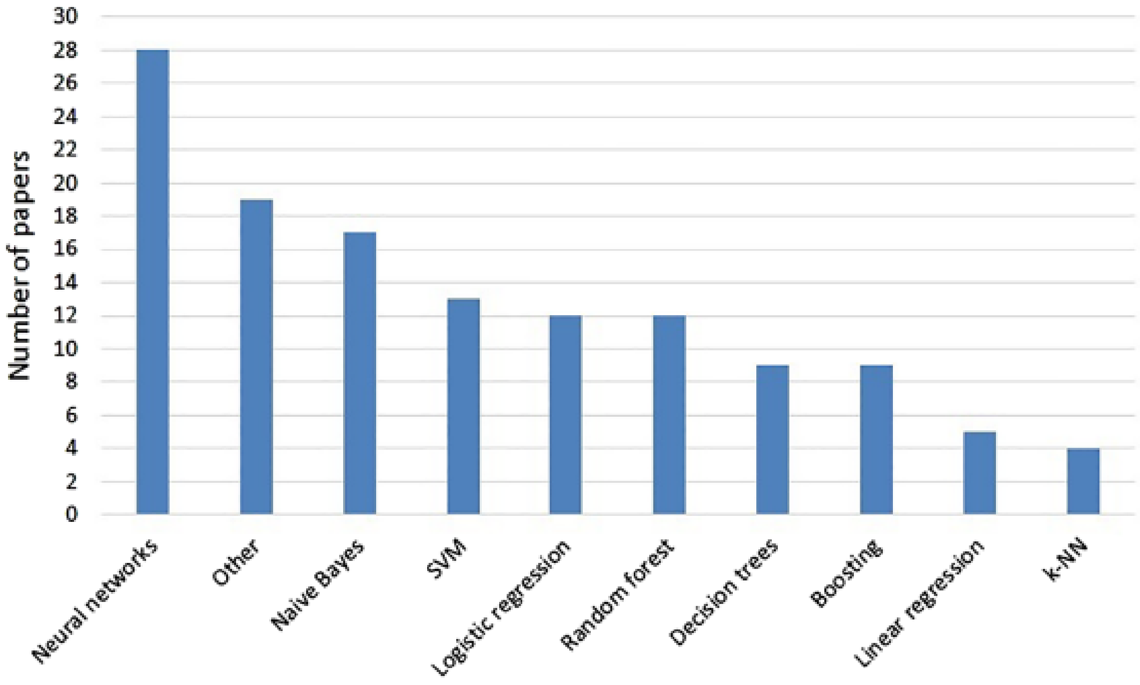

Figure 2.1: Number of papers using a particular ML algorithm group (Horvat and Job, 2020)

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 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 for an over.

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.

2.4 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, $\mathcal{D} = (\mathbf{x}_i, y_i | i = 1, ..., n)$ 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 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

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, lets review the standard linear regression model with Gaussian noise

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

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.

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

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

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

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

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

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

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

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

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

Since the number of obervations 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

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

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

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

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

where we will write the Gaussian process as

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

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 prediciting 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

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

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 !!!

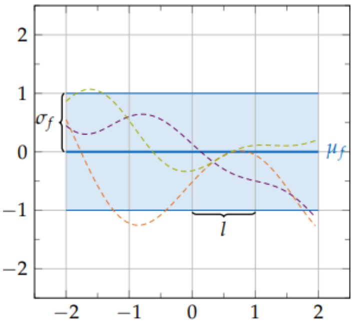

Figure 3.1: SE kernel with $\sigma = 1$ and $l = 1$ (Kaiser, 2017)

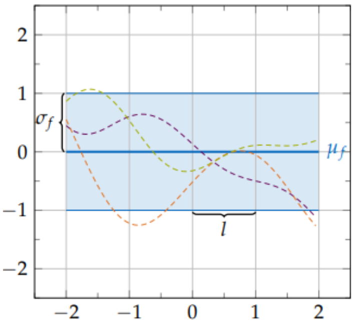

Figure 3.2: SE kernel with $\sigma = \sqrt{2}$ and $l = 1$ (Kaiser, 2017)

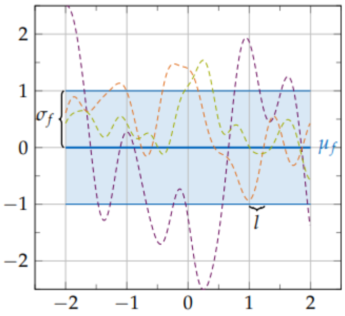

Figure 3.3: SE kernel with $\sigma = 1$ and $l = 0.25$ (Kaiser, 2017)

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 *marginilisation of f*.

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

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

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, and similarly for $K(X, X)$, $K(X_*, X)$ and $K(X_*, X_*)$.

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

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

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

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

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

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

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 B.1.

Algorithm 1 Algorithm for Gaussian process regression

```
Require: $X$ (inputs), $\mathbf{y}$ (targets), $k$ (covariance function), $\sigma_n^2$ (noise level), $X_*$ (test input)
$L \leftarrow cholesky(K + \sigma_n^2 I)$
$\boldsymbol{\alpha} \leftarrow L^T \setminus (L \setminus \mathbf{y})$                          ▷ eq. 3.13
$\bar{f}_* \leftarrow \mathbf{k}_*^T \boldsymbol{\alpha}$                                            ▷ eq. 3.13
$\mathbf{v} \leftarrow L \setminus \mathbf{k}_*$                                                   ▷ eq. 3.14
$\mathbb{V}[f_*] \leftarrow k(X_*, X_*) - \mathbf{v}^T\mathbf{v}$                                    ▷ eq. 3.14
$\log p(\mathbf{y}|X) \leftarrow -\frac{1}{2}\mathbf{y}^T\boldsymbol{\alpha} - \Sigma_i \log L_{ii} - \frac{n}{2}\log 2\pi$    ▷ eq. 3.15
return: $\bar{f}_*$ (mean), $\mathbb{V}[f_*]$ (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* $\mathcal{L}(y_{true}, y_{guess})$ 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 Variational Gaussian Processes

A major drawback on Gaussian processes is their $O(N^3)$ complexity when inversing the covariance matrix K. An approach to reduce this complexity is *Sparse Variational Gaussian Processes* (Hensman, Matthews and Ghahramani, 2014). Spare approximations of GPs are a method of approximating a GP using a $M < N$ set of *inducing points* that can represent the entire dataset, rather than X itself.

3.2 Data Association

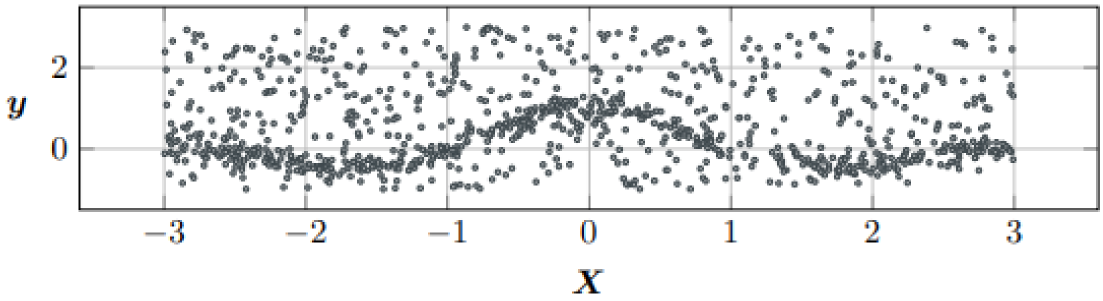

Figure 3.4: A data association problem consisting of two generating processes, one of which is a signl we wish to recover and one is an uncorrelated noise process (Kaiser et al., 2018)

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 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. 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.

For example, in cricket a seam delivery at the top of off stump is generally considered to be a good ball. This knowledge is gathered through anecdotal experience of humans observing many cricket matches and in turn many cricket deliveries. However, it is still entirely possible that a good delivery be hit for a maximum of 6 runs.

Furthermore, a cricket delivery being a good ball is not a simple relationship. For example, lets take a high level look at a seam bowlers length of delivery 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. Anything slighly fuller will be at the batsmens chest, which is considered a poor ball, as it is far easier for the batsmen to get over and hit for boundaries. Equally, slightly fuller 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 again, as a yorker, a ball right at a batsmens feet is considered a very difficult ball to play against. Similar examples can be made with the width of a ball, its swing, its spin, and many other features.

Equally, a "good" or "bad" ball is obviously subjective depending on the match state and what the bowling team is trying to achieve. Typically in an IPL match, the goal of the bowling team is to reduce the oppositions runs scored. The best balls for this requirement, are not the same as the best balls to take a wicket. Equally, there is an idea in cricket of "setting a batsmen up". This idea is having the previous deliveries set the batsmen up to get out to the next ball. For example, a short ball to throw the batsmen off, followed by a yorker right at the stumps.

From theses examples, 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.

3.2.2 Data Association with Gaussian Processes

(Kaiser et al., 2018)

3.2.3 Modulated Scalable Gaussian Processes

(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://robblumberg.github.io/exp-goals/.

Echo, L., 2019. The unknown liverpool backroom analyst who convinced jurgen 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.espncricinfo.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.

Hensman, J., Matthews, A. and Ghahramani, Z., 2014. Scalable variational gaussian process classification.

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-YGsS9bhdiOkYkurvCmBCdZpbAx-MV4VF806rtp_h1CXJlCCOpg4hFFknpmc9dAb3wBAVu7NZ_FOBQ.

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://singlearity.com/static/assets/sloan-singlearity.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.

Wikipedia, 2023a. [Online]. Last accessed 18 April 2023. Available from: https://en.wikipedia.org/wiki/Cricket#.

Wikipedia, 2023b. [Online]. Last accessed 18 April 2023. Available from: https://en.wikipedia.org/wiki/Leg_before_wicket#.

Yi, W., 2019. Understanding gaussian process, the socratic way [Online]. Available from: https://towardsdatascience.com/understanding-gaussian-process-the-socratic-way-ba02369d804.

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 T20 Cricket

There are varying formats of cricket and the focus in this project is the T20 format (20 over cricket) that is played in the IPL, see section 1.2.1. T20 cricket gives the first side 20 overs to score as many runs as they can, 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 they get all out before the target total, or use all 20 overs before reaching the target, they lose. 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

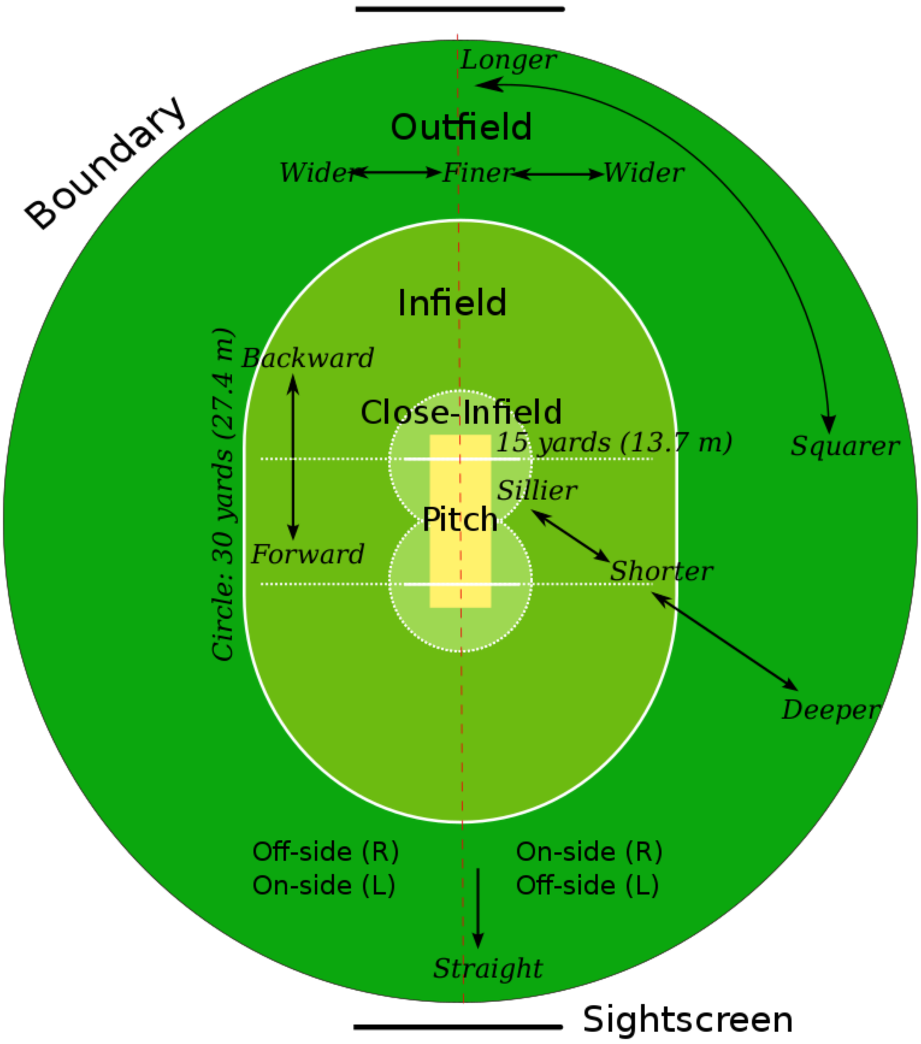

Figure A.1: A typical cricket field (Wikipedia, 2023a)

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.

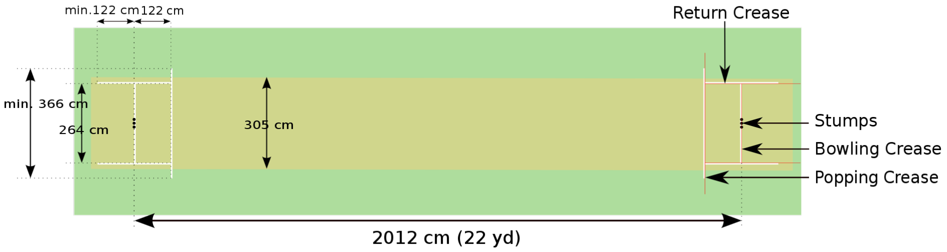

Figure A.2: A typical cricket pitch (Wikipedia, 2023a)

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.

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 project I will only explain the most common dismissals.

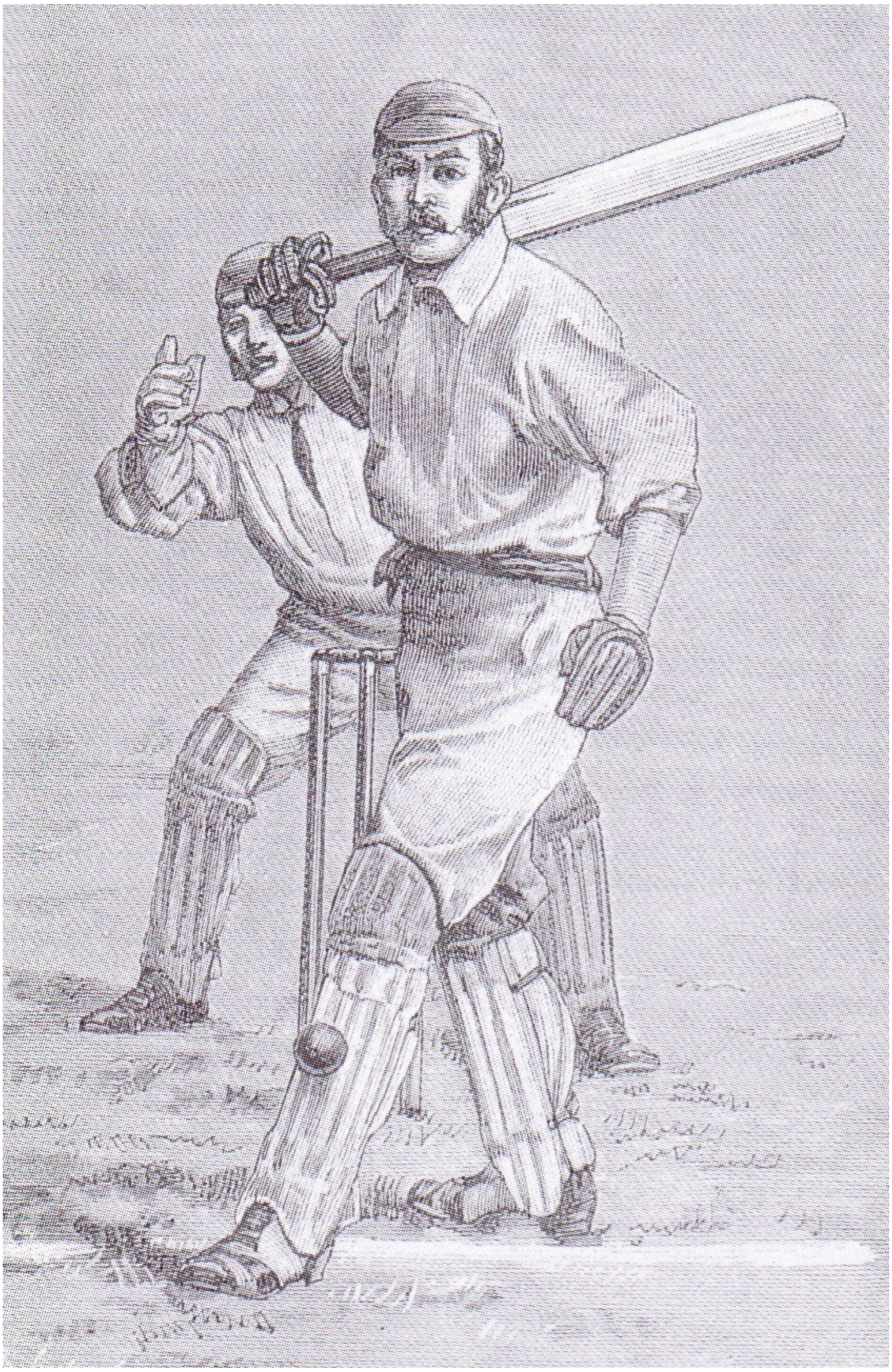

Figure A.3: A clear example of leg before wicket (lbw) (Wikipedia, 2023b)

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

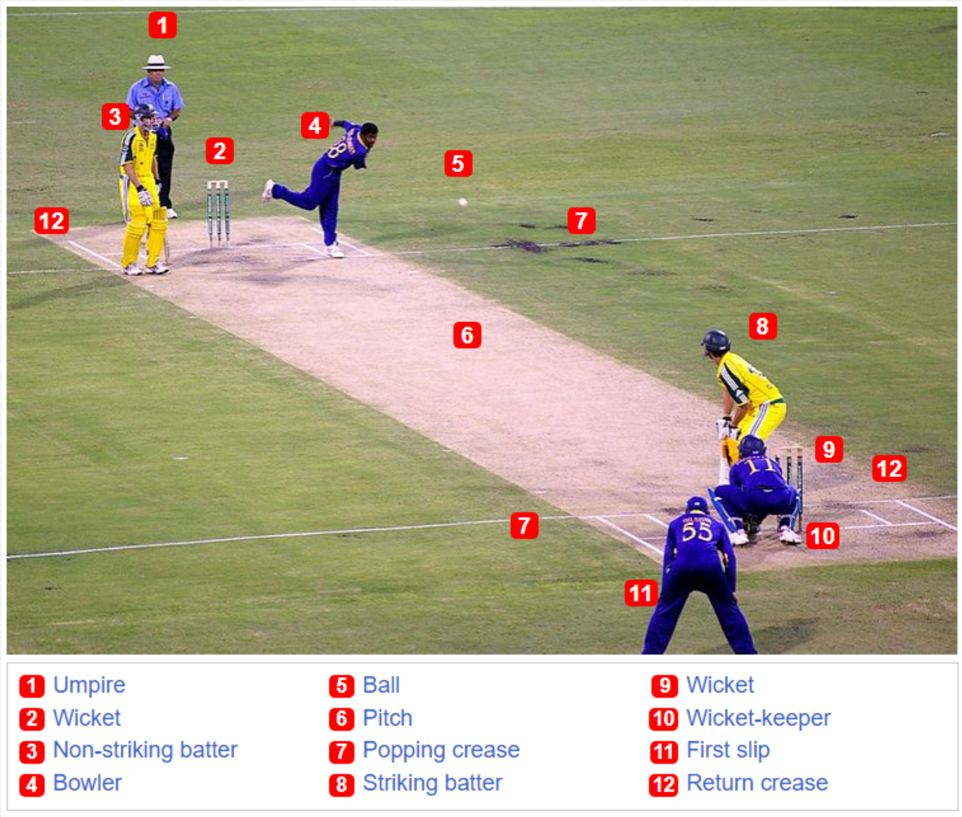

Figure A.4: An example of a ball being bowled and the components in play (Wikipedia, 2023a)

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 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

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");
}
```