University of BRISTOL

Modelling outcomes of football matches using the Bradley-Terry Model

Matthew Barrett

Supervised by Dr Anthony Lee
Level M/7
40 Credit Points

May 17, 2021

Acknowledgement of Sources

Acknowledgement of Sources

For all ideas taken from other sources (books, articles, internet), the source of the ideas is mentioned in the main text and fully referenced at the end of the report.

All material which is quoted essentially word-for-word from other sources is given in quotation marks and referenced.

Pictures and diagrams copied from the internet or other sources are labelled with a reference to the web page or book, article etc.

Signed MBarrett

Date 17/05/21

Abstract

The global sports betting industry has seen huge growth in recent years, highlighting the demand for accurately forecasting sports results. This paper will look at predicting the outcomes of matches in the Premier league, the top league in English football, using the Bradley-Terry model. We will look at adding various extensions to the original Bradley-Terry model such as how to incorporate the possibility of draws and how to incorporate the home advantage effect, which is a well-known phenomenon in sports whereby a team is more likely to win when playing at their own facility. We will consider implementing both a frequentist technique of maximum likelihood estimation and a Bayesian technique of using Markov chain Monte Carlo methods to infer the parameters of the model. Throughout the paper the main measure of the success of a model is simply the percentage of matches that the model correctly predicts. We find that the two approaches of inference, whilst differing philosophically, result in a very similar quality of predictions. In particular, we find that a Bradley-Terry model with extensions to incorporate draws and the home advantage effect provides the most accurate forecasts.

Contents

1 Introduction

1.1 Literature review

In this paper we will be looking at how to use the Bradley-Terry model to predict the outcomes of football matches. The Bradley-Terry model was first studied by Zermelo (1929) before being rediscovered by Bradley and Terry (1952). The model states that the probability of object i being preferred to object j is given by

$$\frac{\pi_i}{\pi_i + \pi_j}$$

where $\pi_i \geq 0$ is a positive score assigned to object i. In the context of football, the objects are the teams and we will refer to the scores as the strengths of the teams. We demonstrate the versatility of the Bradley-Terry model by looking at a few applications of it. Selby and Firth (n.d.) look at ranking different academic fields in terms of how much they influence other fields. In this scenario the objects are the academic fields, and for example statistics is preferred to medicine if an article from a medicine journal cites an article from a statistics journal. Looking at 20 million citations from various scientific fields, they find that statistics is the field with the most interdisciplinary influence. The Bradley-Terry model is sometimes used in taste-testing where subjects repeatedly compare the tastes of two items from some set of products, in order to rank the products from best to worst taste. Lukas (1991) uses the Bradley-Terry model to rank different champagnes in terms of their taste qualities. Matthews and Morris (1995) have patients repeatedly receive two injections, with each of them being administered in slightly different ways. The patients are then asked to judge which felt more painful, producing a series of paired comparisons and allowing the experimenters to infer which of the four administration techniques is preferred by the patients.

Independent works by Plackett (1975) and Luce (1959) have given rise to an extension of the Bradley-Terry model which allows for the comparison of more than just two objects, known as the Plackett-Luce model. The Plackett-Luce model has found applications in forecasting the outcomes of Formula One races (Henderson and Kirrane, 2017), since a Formula One race sees 20 drivers compete against each other. The Plackett-Luce model has also been useful for document ranking, which is the problem behind search engines where the engine is given a query and must try to return the most relevant results first. Kuo et al. (2009) explores the problem of document ranking through the use of the Plackett-Luce model and a neural network. Now having seen applications of the Bradley-Terry model to so many other academic fields, it is perhaps not such a surprise that Selby and Firth (n.d.) found statistics to be such an influential field!

In recent years there has been substantial interest in applying statistical models to the predictions of sporting events. It is apparent why when we consider that the global sports betting industry reached a market size of 203 billion U.S. dollars in 2020 (Lock, 2021). In the context of football it is understood that Maher (1982) was the first instance of using a model to make predictions about future games. Maher (1982) assumes the number of goals scored by the teams are drawn from a Poisson distribution, where the mean of the Poisson varies depending on the attacking and defensive qualities of the given teams. Building on this model, Dixon and Coles (1997) make adjustments to correct for the fact that the Poisson model gives poor predictions of low scoring games. They also incorporate a weighting function to downweight older games which are assumed to be less important in the estimation of a team's current ability.

In order to apply the Bradley-Terry model to football data we must find a way to incorporate draws into the model. To allow for the possibility of draws Rao and Kupper (1967) propose an extension whereby a draw is declared if the scores of the objects are within some threshold of one another. Alternatively, Davidson (1970) suggests that the probability of a draw be proportional to the geometric mean of the probabilities that either object is preferred. Additionally, it is well documented that in football, teams tend to perform better when playing at home than when playing away (Clarke and Norman, 1995). Although, Tilp and Thaller (2020) found that the Covid-19 lockdown actually led to a home disadvantage in Germany's top football league, since fans were no longer allowed to attend games. We won't worry about this potential complication however, as we will apply our models to data from before the Covid-19 pandemic when there were no restrictions on fans attending games. To take into account the effect of the home advantage Agresti (2003) proposes that the probability of team i beating team j should be given by

$$P(\text{i beats j}) = \begin{cases} \frac{\theta\pi_i}{\theta\pi_i+\pi_j}, & \text{if i is home} \\ \frac{\pi_i}{\pi_i+\theta\pi_j}, & \text{if j is home} \end{cases}$$

where $\theta > 0$ measures the magnitude of the home advantage (or disadvantage). Tsokos et al. (2019) assess the potential of both the Poisson based model and the Bradley-Terry model with respect to how well they predict future outcomes, and found that they exhibit similar predictive performance. Amongst their different Bradley-Terry model specifications, they report the best performing to be one in which they model the log strengths of the teams to depend linearly on unknown smooth functions of some explanatory variables, such as the team's goal difference before the current match. Such a technique is known as a generalised additive model, details for which can be found in Wood (2006). Finally, Constantinou et al. (2012) use a Bayesian network to predict match outcomes, and by incorporating the subjective opinion of an expert into an objective model they were able to improve the predictive accuracy and formulate a profitable betting strategy.

When it comes to inferring the parameters of our models it turns out that there already exist packages in R that can do this for us. Turner and Firth (2012) create an R package which implements the maximum likelihood estimation of the parameters of the Bradley-Terry model and extensions of the model such as allowing for order effects (home advantage). Alternatively, to perform inference by using Markov chain Monte Carlo methods one can use the software provided by Stan Development Team (2020), which obtains samples from the posterior distribution by making use mainly of the No-U-Turn sampler (Hoffman and Gelman, 2014). We have set out however to present and compare the methods of maximum likelihood estimation and Markov chain Monte Carlo methods. When comparing the techniques the ease of implementation is an important factor when considering which of the methods is better. We will therefore implement both techniques ourselves rather than using any R packages, to allow us to compare the methods not only in terms of predictive performance but also in terms of ease of implementation and computational efficiency.

1.2 Overview

Section two of this paper offers a justification for the structure of the Bradley-Terry model. We also look at an extension to incorporate draws, again justifying the model formulation and checking its validity using some Premier League data.

In section three we look into our first method of inference, maximum likelihood estimation. We show the conditions necessary for the maximum likelihood estimate to exist and to be unique for the original Bradley-Terry model and for the extended model with draws. Some theory of maximum likelihood estimation is then explored to allow us to find the standard errors of our estimates. Finally, we look at how well the model is able to predict the outcomes of football matches when we infer the parameters by maximum likelihood estimation.

Considering an alternative method of inference in section four, we set prior distributions on the parameters to encode our beliefs about their values and find the mean of the posterior distribution. In particular, we will use the Metropolis-within-Gibbs algorithm (Metropolis et al., 1953) to sample from the posterior distribution and approximate the posterior mean with these samples. Then with the posterior mean as our parameter estimates we assess again the predictive accuracy of the model.

In the last section we consider how we can take into account explanatory variables, such as location (home/away) and team form, to try to improve the accuracy of our predictions. In general we find that our best model correctly predicts around 50-60% of games. The predictive accuracies are relatively low, which we believe to be due to two main reasons: the unpredictable nature of football and the model oversimplifying the reality of how the data is generated.

To understand why football is so unpredictable in nature, we first notice the frequency with which draws occur in the game. For instance in the data we look at from the 16/17 Premier League season, draws account for 22.1% of all outcomes. Naturally, predicting which of three outcomes will occur is more challenging than predicting which of two outcomes will occur. Hence football is more unpredictable than games that don't allow draws, such as tennis, or games which encounter draws at a much lesser rate, such as rugby or American football. Additionally, football is a game which sees a relatively low number of scoring opportunities. To see why this may be problematic let us imagine a game which instead generally has a large number of scoring opportunities. Clearly a better team will score more of its opportunities i.e. they have a higher probability of scoring an opportunity. Since the number of scoring opportunities is large and appealing to the law of large numbers, it is almost certain that the better team will win the game. However, given the low scoring nature of football the opposite is true, and it seems that it may not always be the case that the better team wins. This is summarised nicely by Reep and Benjamin (1968), "chance does dominate the game".

Secondly, it seems that the model doesn't capture the true complexity of how the data is generated. More specifically, the Bradley-Terry model satisfies transitivity of preferences i.e. if A is preferred to B and B is preferred to C, then A is preferred to C. However, football is full of violations of this rule. This is because teams can't be summarised by a single number representing their strength, but rather each team differs along multiple dimensions, including their fitness, the strength of their midfield, the strength of their substitutes etc. Therefore we lose information by assuming that teams can be summarised by a single value, but of course this is a caveat associated with most statistical models.

2 Model

2.1 Motivation

We want to justify the way the Bradley-Terry model is formulated. In particular, we seek to find an expression for the probability that some team, A, beats another team, B, in terms of their respective win rates. Let $P_{A,B}$ be the probability that team A beats team B, and P_A, P_B be the true win rates of A and B respectively. To estimate this probability, imagine that A and B are playing a game where A gets 1 with probability P_A or 0 with probability $1 - P_A$, and similarly B gets 1 with probability P_B and 0 with probability $1 - P_B$. The one with the higher number wins the game, and if the numbers are equal we play again until a winner is determined. Note that we can think of the teams getting 1 to mean that the team played well, and similarly getting a 0 implies they didn't play well.

Therefore the probability that A wins on the first go is the probability that A gets 1 and B gets 0, which is $P_A(1 - P_B)$. Similarly, the probability that A wins in the second round is the probability that there's a draw in the first round and A wins the second, given by

$$[P_AP_B + (1 - P_A)(1 - P_B)]P_A(1 - P_B) = (1 - P_A - P_B + 2P_AP_B)P_A(1 - P_B).$$

Following the pattern, we have that the probability of A winning in the n^{th} round is the probability of draws in the first $n - 1$ rounds and A wins in the n^{th} round, given by $(1 - P_A - P_B + 2P_AP_B)^{n-1}P_A(1 - P_B)$. Summing over n and using the sum of a geometric series, we find that

$$\begin{aligned}
P_{A,B} &= \sum_{n=1}^{\infty}(1 - P_A - P_B + 2P_AP_B)^{n-1}P_A(1 - P_B) \\
&= P_A(1 - P_B)\sum_{n=0}^{\infty}(1 - P_A - P_B + 2P_AP_B)^n \\
&= P_A(1 - P_B)\frac{1}{1 - (1 - P_A - P_B + 2P_AP_B)} \\
&= \frac{P_A(1 - P_B)}{P_A + P_B - 2P_AP_B}
\end{aligned}$$

Note that we can only sum the geometric series if $|1-P_A-P_B+2P_AP_B| < 1$, which holds as long as P_A and P_B aren't both equal to 0 or 1.

We can use this formula to recover the Bradley-Terry model. If an event occurs with probability p, then the odds of that same event are given by $\frac{p}{1-p}$, and now let us consider instead the odds of A beating B:

$$\begin{aligned}
\text{Odds}_{A,B} &= \frac{P_{A,B}}{1 - P_{A,B}} \\
&= \frac{P_A(1 - P_B)}{P_A + P_B - 2P_AP_B} \div \left(1 - \frac{P_A(1 - P_B)}{P_A + P_B - 2P_AP_B}\right) \\
&= \frac{P_A(1 - P_B)}{P_A + P_B - 2P_AP_B} \div \frac{P_A + P_B - 2P_AP_B - P_A + P_AP_B}{P_A + P_B - 2P_AP_B} \\
&= \frac{P_A(1 - P_B)}{P_A + P_B - 2P_AP_B} \times \frac{P_A + P_B - 2P_AP_B}{P_B - P_AP_B} \\
&= \frac{P_A(1 - P_B)}{P_B - P_AP_B} \\
&= \frac{P_A}{1 - P_A}\frac{1 - P_B}{P_B}
\end{aligned}$$

Taking natural logarithms we have

$$\begin{aligned}
\log(\text{Odds}_{A,B}) &= \log\left(\frac{P_A}{1 - P_A}\right) + \log\left(\frac{1 - P_B}{P_B}\right) \\
&= \log\left(\frac{P_A}{1 - P_A}\right) - \log\left(\frac{P_B}{1 - P_B}\right)
\end{aligned}$$

and we see that although the probability of A beating B cannot be represented as a linear combination of the team 'strengths', the log odds can be. Taking the log strength of team i to be $\lambda_i = \log\left(\frac{P_i}{1-P_i}\right)$, putting it into the above equation and converting back to probabilities, we obtain the

Bradley-Terry model:

$$\begin{aligned}\log(\text{Odds}_{A,B}) = \lambda_A - \lambda_B &\iff \text{Odds}_{A,B} = e^{\lambda_A - \lambda_B}\\ &\iff P_{A,B} = \frac{e^{\lambda_A - \lambda_B}}{1 + e^{\lambda_A - \lambda_B}}\\ &\iff P_{A,B} = \frac{e^{\lambda_A}}{e^{\lambda_B} + e^{\lambda_A}}\end{aligned}$$

where the second equivalence is because the probability, p, of an event written in terms of the odds of that event is $p = \frac{\text{odds}}{1+\text{odds}}$. To confirm the model is a somewhat sensible choice we state a few of its desirable properties

- $P_{A,B} + P_{B,A} = \frac{e^{\lambda_A}}{e^{\lambda_B}+e^{\lambda_A}} + \frac{e^{\lambda_B}}{e^{\lambda_B}+e^{\lambda_A}} = 1$ (outcome probabilities sum to 1)
- If $\lambda_A = \lambda_B$, then $P_{A,B} = P_{B,A} = 0.5$ (Equally strong teams have equal chance of winning)
- As $\lambda_A \to \infty, P_{A,B} \to 1$ (infinitely strong team always wins)
- As $\lambda_A \to -\infty, P_{A,B} \to 0$ (infinitely weak team always loses)

2.2 Bradley-Terry model and incorporating ties

In our context, the Bradley-Terry model assigns probabilities as follows. For a paired comparison between team i and team j, we have (i>j means team i beats team j)

$$\begin{aligned}P(i > j) &= \frac{\pi_i}{\pi_i + \pi_j}\\ P(j > i) &= \frac{\pi_j}{\pi_i + \pi_j}\end{aligned}$$

where $\pi_i = e^{\lambda_i}$, and π_i is a positive real value assigned to team i which can be thought of as the 'strength' of team i. Since we plan to apply this model to the Premier League, we need to address the fact that our model isn't yet able to assign a probability to the outcome of a draw, which in the Premier League happens about a quarter of the time. To gain some insight into how the model would perform anyway, we consider all results from the 2016/17 season that didn't end in a draw. For each decided game we assign the appropriate probability given by the above model, at which point we can construct the likelihood function and find the maximum likelihood estimate of the strength of each team (exact method is outlined in section 3.2). Now with a theoretical strength associated with each team we compare the order that the model ranks the teams in with the true rankings at the end of the 16/17 season in table 1.

As we can see in table 1, the model rankings differ from the actual rankings in quite a few places, motivating us to extend the model to incorporate

Position	Model Rankings	True Rankings
1	Tottenham	Chelsea
2	Chelsea	Tottenham
3	Man United	Man City
4	Man City	Liverpool
5	Liverpool	Arsenal
6	Arsenal	Man united
7	Everton	Everton
8	West Brom	Southampton
9	West Ham	Bournemouth
10	Southampton	West Brom
11	Leicester	West Ham
12	Bournemouth	Leicester
13	Stoke	Stoke
14	Watford	Crystal Palace
15	Burnley	Swansea
16	Crystal Palace	Burnley
17	Swansea	Watford
18	Hull	Hull
19	Middlesbrough	Middlesbrough
20	Sunderland	Sunderland

Table 1: Rankings implied by the original Bradley-Terry model vs the true rankings, 16/17

draws. If we can assign an appropriate probability to the outcome of a draw, then we can use all outcomes in calculating the strengths, as opposed to ignoring any game ending in a draw. We expect that such a model will find a ranking of the teams that more closely resembles the true rankings. Davidson (1970) suggests that the probability of no preference (draw) between i and j be proportional to the geometric mean of the probabilities of preference for i and j i.e

$$P(\text{i,j draw}) = P(i \leftrightarrow j) = \nu\sqrt{P(i > j)P(j > i)}$$

where ν is a constant of proportionality. The use of the geometric mean, Davidson states, is implied by the fact that the strengths can be represented by the values $\log(\pi_1), ..., \log(\pi_{20})$ on a linear scale. Using the above we can derive a new model which specifies a probability for each outcome that a football match can take. We have that the probabilities of team i beating, drawing and losing to team j are respectively given by

$$P(i > j) = \frac{\pi_i}{\pi_i + \pi_j + \nu\sqrt{\pi_i \pi_j}}$$

$$P(i \leftrightarrow j) = \frac{\nu\sqrt{\pi_i \pi_j}}{\pi_i + \pi_j + \nu\sqrt{\pi_i \pi_j}}$$

$$P(j > i) = \frac{\pi_j}{\pi_i + \pi_j + \nu\sqrt{\pi_i \pi_j}}$$

where $\pi_i = e^{\lambda_i}$ and $\nu \geq 0$ is a constant to be estimated from the data. The larger ν is the larger the probability of a draw is, so ν is determined by the prevalence of draws in the data. The model has the nice property that if there are no draws then $\nu = 0$, since the probability of a draw is 0, in which case the above model simply becomes the original Bradley-Terry model. We also note that

$$\begin{aligned} P(i > j|\text{not draw}) &= \frac{P(i > j, \text{not draw})}{P(\text{not draw})} \\ &= \frac{P(i > j)}{P(i > j) + P(j > i)} \\ &= \frac{\pi_i}{\pi_i + \pi_j + \nu\sqrt{\pi_i \pi_j}} \div \frac{\pi_i + \pi_j}{\pi_i + \pi_j + \nu\sqrt{\pi_i \pi_j}} \\ &= \frac{\pi_i}{\pi_i + \pi_j} \end{aligned}$$

Hence conditional on the outcome not being a draw, we again obtain the original Bradley-Terry model. Also the maximum value for $P(i \leftrightarrow j)$ will be obtained when $\nu\sqrt{\pi_i \pi_j}$ is maximised. The arithmetic-geometric mean inequality tells us that

$$\sqrt{\pi_i \pi_j} \leq \frac{\pi_i + \pi_j}{2}$$

with equality if and only if $\pi_i = \pi_j$. So we see that $\nu\sqrt{\pi_i \pi_j}$ is maximised when $\pi_i = \pi_j$, or rather, the probability of a draw is greatest when the two teams have equal strengths, as we would expect.

Now with our updated model we estimate again the strengths of all the teams by maximum likelihood estimation, rank the teams from highest strength to lowest strength, and compare the ordering to the actual final league table. The results of this procedure are shown in table 2.

Comparing table 2 to table 1 it's apparent that the extended model is much closer to ranking the teams correctly. Table 2 only seems to mix up a few of the teams in the middle of the table, but looking at the actual 16/17 final table all of the mid-table teams have a very similar number of points. Hence the fact that the model mixes some of them up is not much cause for concern.

Position	Model Rankings	True Rankings
1	Chelsea	Chelsea
2	Tottenham	Tottenham
3	Man City	Man City
4	Liverpool	Liverpool
5	Arsenal	Arsenal
6	Man United	Man united
7	Everton	Everton
8	Bournemouth	Southampton
9	Southampton	Bournemouth
10	West Brom	West Brom
11	Stoke	West Ham
12	West Ham	Leicester
13	Leicester	Stoke
14	Swansea	Crystal Palace
15	Watford	Swansea
16	Burnley	Burnley
17	Crystal Palace	Watford
18	Hull	Hull
19	Middlesbrough	Middlesbrough
20	Sunderland	Sunderland

Table 2: Rankings implied by the extended Bradley-Terry model vs the true rankings, 16/17

3 Frequentist inference

3.1 Maximum likelihood estimation

Now we have a model which can specify a probability to each outcome of a game, in terms of the log strengths, $\lambda_1, ..., \lambda_{20}$, and a constant ν controlling the prevalence of draws in the case of the extended model. Note that λ_i is the log strength of team i, and we arbitrarily assign each team a number between 1 and 20. We want to estimate these parameters from our data, so to accomplish this we first construct the likelihood function. This is basically the probability, under our model, of observing the data that we have. As an example, consider that we have the following data

Home team	Score	Away team
Burnley (Team 1)	0 - 1	Swansea (Team 2)
Everton (Team 3)	1 - 1	Tottenham (Team 4)

Table 3: Theoretical match outcomes

and noticing that Swansea beat Burnley and Everton draw against Tottenham, the likelihood function given the data, call it x, is given by

$$\begin{aligned}
L(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \nu; x) &= f(x; \lambda_1, \lambda_2, \lambda_3, \lambda_4, \nu) \\
&= P(2 > 1, 3 \leftrightarrow 4; \lambda_1, \lambda_2, \lambda_3, \lambda_4, \nu) \\
&= P(2 > 1; \lambda_1, \lambda_2, \nu) P(3 \leftrightarrow 4; \lambda_3, \lambda_4, \nu) \\
&= \frac{e^{\lambda_2}}{e^{\lambda_2} + e^{\lambda_1} + \nu\sqrt{e^{\lambda_1} e^{\lambda_2}}} \frac{\nu\sqrt{e^{\lambda_3} e^{\lambda_4}}}{e^{\lambda_3} + e^{\lambda_4} + \nu\sqrt{e^{\lambda_3} e^{\lambda_4}}}
\end{aligned}$$

where $f(x; \cdot)$ is the probability mass function given by the extended model and the third equality is obtained by making the reasonable assumption that the two games are independent of each other i.e. the fact that Swansea beat Burnley probably has no meaningful effect on the Everton vs. Tottenham game. Note that we are interested in finding the log strength terms, λ_i. Since the strength terms, π_i, are positive it is easier to find the log strengths which can take any real value, meaning we don't need to worry about constraining the parameters. More generally, still assuming independence, with data $x = (x_1, ..., x_n)$ where x_i is the outcome of game i and θ is the vector of parameters, the likelihood function is given by

$$L(\theta; x) = f(x; \theta) = P(x_1, ..., x_n; \theta) = \prod_{i=1}^{n} P(x_i; \theta)$$

Given that we are in a frequentist setting, we view the parameters of our model as unknown but fixed, and we define our estimates of these parameters to be equal to

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L(\theta; x)$$

where $\Theta = \mathbb{R}^{20}$ for the original Bradley-Terry model and $\Theta = \mathbb{R}^{20} \times \mathbb{R}^{+}$ is the parameter space in the extended model. $\hat{\theta}$ is known as the maximum likelihood estimate (MLE). The aim of this procedure is to find the values of the parameters for which the likelihood function attains its maximum value, or more intuitively, it tells us the values of the parameters which make us most likely to observe the data that we have. To justify the use of the MLE, we note that assuming some regularity conditions and that the data is generated by $f(\cdot; \theta_0)$, then the MLE is a consistent estimator. This means that as the sample size n tends to ∞, $\hat{\theta}$ converges to θ_0 in probability, where θ_0 are the true values of the parameters. Of course, the data isn't actually generated by $f(\cdot; \theta_0)$, it is just our best attempt to model how the data is generated. However consistency is still a favourable theoretical property for the estimator to have.

Before we can compute the MLE we have some problems we must address. Firstly, if we take a look at the form of the likelihood function, we notice that it is the product of n probabilities, with n the number of observed games. A season in the Premier League sees 380 games played, meaning the likelihood function when taking in to account all games from a season will be the product of 380 probabilities. Due to the complexity of the problem, we rely on numerical optimisation methods in R to find the MLE. It could therefore easily be the case that the value of the likelihood function is even smaller than the precision of R, which means R will just report the value to be 0 (e.g. R reports 0.1^{380} to be 0, which is wrong). This would be problematic since we would be telling the computer to find the maximum of a function which is always 0- the MLE would trivially be any combination of values of the parameters, which would not be useful. To solve this, we consider instead using the natural logarithm of the likelihood function, we have

$$\ell(\theta; x) = \log(L(\theta; x)) = \log\left(\prod_{i=1}^{n} P(x_i; \theta)\right) = \sum_{i=1}^{n} \log(P(x_i; \theta))$$

and our expression now takes the form of the sum of log-probabilities.

To demonstrate that taking logs is an appropriate thing to do, we will show that if $\hat{\theta}$ uniquely maximises the likelihood function, then $\hat{\theta}$ also uniquely maximises the log likelihood function. Let $\hat{\theta}$ be the unique maximiser of the likelihood function, then for any $\theta \neq \hat{\theta}$ we have

$$L(\hat{\theta}; x) > L(\theta; x).$$

Since the logarithm is a monotonically increasing function, taking logs of both sides yields

$$\log(L(\hat{\theta}; x)) > \log(L(\theta; x)) \iff \ell(\hat{\theta}; x) > \ell(\theta; x)$$

showing that $\hat{\theta}$ does indeed also uniquely maximise the log likelihood, so it makes sense for us to focus instead on maximising the log likelihood.

To see why this may be beneficial, imagine we have some parameters θ' such that with these values the probability of all the observed outcomes is 0.1 i.e. $P(x_i; \theta') = 0.1 \; \forall i$. Suppose also that we have parameters θ'' which make the probability of all outcomes 0.11. Computing the values of the likelihood function for θ' and θ'', in R, gives $L(\theta'; x) = 0.1^{380} = 0$ and $L(\theta''; x) = 0.11^{380} = 0$. Since we are trying to find the values of the parameters which make the observed data most likely, clearly θ'' is a better choice than θ', however due to the likelihood function taking such small values R isn't able to distinguish between θ' and θ''. Now for the same parameters the values of the log likelihood function, given by R, are $\ell(\theta'; x) = 380 \times \log(0.1) = -874.9823$ and $\ell(\theta''; x) = 380 \times \log(0.11) = -838.7645$. So in this case R can distinguish between θ' and θ'', and in particular $\ell(\theta'; x) < \ell(\theta''; x)$ implies that θ'' are the preferred parameters here, which is exactly what we want.

An additional benefit of using the log likelihood is that it is a much simpler function to differentiate compared to the likelihood function. This will be helpful when we come to find the MLE which we defined to be the vector of values $\hat{\theta}$ which maximises $L(\theta; x)$. To find this value we need to find the stationary point of $L(\theta; x)$, and to do this we would need to find the solution to the equation $\nabla L(\theta; x) = 0$. However since the likelihood function is a product of (up to 380) terms, it will become increasingly burdensome to find the partial derivatives analytically. So, as justified above, we can equivalently find the maximum of the log likelihood function i.e. the solution to $\nabla\ell(\theta;x) = 0$. Finding the partial derivatives of the log likelihood proves much easier since it takes the form of a sum and so with $k \in \theta$, we have

$$\frac{\partial \ell(\theta;x)}{\partial k} = \frac{\partial}{\partial k}\sum_{i=1}^{n} \log(P(x_i;\theta)) = \sum_{i=1}^{n} \frac{\partial}{\partial k} \log(P(x_i;\theta)).$$

Hence to find a partial derivative of $\ell(\theta;x)$ we just need to find the partial derivative of the log probability associated with each game and sum them up. In a sentence, finding the derivative of the log likelihood function is simpler than finding the derivative of the likelihood function due to the fact that for functions f,g $(f+g)' = f' + g'$ but in general $(fg)' \neq f'g'$. This will be useful when it comes to approximating the MLE in R, as we will be able to find the derivative analytically which we will see speeds up the computation significantly. Now we can see that simply taking logs yields a more numerically stable and efficient way of finding the MLE, so from now on we consider the equivalent problem of finding

$$\hat{\theta} = \arg\max_{\theta\in\Theta} \ell(\theta;x).$$

The final problem we have with the model is one of identifiability. Let the log strength terms be equal to $\lambda_i + c$ for each i, where c is a constant. Then in the case of the original Bradley-Terry model we have

$$P(i > j) = \frac{e^{\lambda_i + c}}{e^{\lambda_i + c} + e^{\lambda_j + c}} = \frac{e^c e^{\lambda_i}}{e^c(e^{\lambda_i} + e^{\lambda_j})} = \frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j}}$$

and in the case of the extended model we have

$$P(i > j) = \frac{e^{\lambda_i + c}}{e^{\lambda_i + c} + e^{\lambda_j + c} + \nu\sqrt{e^{\lambda_i + c}e^{\lambda_j + c}}} = \frac{e^c e^{\lambda_i}}{e^c e^{\lambda_i} + e^c e^{\lambda_j} + \nu\sqrt{e^{2c}}\sqrt{e^{\lambda_i}e^{\lambda_j}}}$$

$$= \frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i}e^{\lambda_j}}}$$

$$P(i \leftrightarrow j) = \frac{\nu\sqrt{e^{\lambda_i + c}e^{\lambda_j + c}}}{e^{\lambda_i + c} + e^{\lambda_j + c} + \nu\sqrt{e^{\lambda_i + c}e^{\lambda_j + c}}} = \frac{\nu\sqrt{e^{2c}}\sqrt{e^{\lambda_i}e^{\lambda_j}}}{e^c e^{\lambda_i} + e^c e^{\lambda_j} + \nu\sqrt{e^{2c}}\sqrt{e^{\lambda_i}e^{\lambda_j}}}$$

$$= \frac{\nu\sqrt{e^{\lambda_i}e^{\lambda_j}}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i}e^{\lambda_j}}}.$$

We can see that in both cases the value of the constant c is irrelevant and the probability given by the model will be the same if for each i the log strength of team i is equal to λ_i or if the log strength of team i is $\lambda_i + c$. We say that the log strengths are unique only up to an additive constant, or equivalently, the strengths are unique only up to a multiplicative constant. To solve this we constrain the strengths such that

$$\sum_{i=1}^{20} \pi_i = 1 \iff \sum_{i=1}^{20} e^{\lambda_i} = 1.$$

This constraint ensures that any value of the log likelihood is determined uniquely by some set of strengths and as a result the MLE with respect to the above constraint, if it exists, will be unique. Now that we have constrained the problem we need to appropriately update the parameter space. With $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_{20})$, in the case of the original Bradley-Terry model we have parameter space $\Theta = \{\boldsymbol{\lambda} \in \mathbb{R}^{20} : \sum_{i=1}^{20} e^{\lambda_i} = 1\}$ and for the model incorporating ties we have $\Theta = \{\boldsymbol{\lambda} \in \mathbb{R}^{20} : \sum_{i=1}^{20} e^{\lambda_i} = 1\} \times \mathbb{R}^+$.

3.2 Inference for the original Bradley-Terry model

In this section we will demonstrate how to perform maximum likelihood estimation for the original Bradley-Terry model. Although we will not consider any predictions given by this model, it will serve as a slightly simpler introduction to some of the ideas and we will show how table 1 from section 2.2 was obtained. Recall that the original Bradley-Terry model assigns probabilities in the following way

$$P(i > j) = \frac{\pi_i}{\pi_i + \pi_j} = \frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j}}.$$

So with $\theta = (\lambda_1, ..., \lambda_{20}) \in \mathbb{R}^{20}$ and $x = (x_1, ..., x_n)$ we can write the log likelihood function as

$$\ell(\theta; x) = \sum_{k=1}^{n} \log(P(x_k; \theta)) = \sum_{i,j} w_{ij} \log(P(i > j))$$

where w_{ij} is the number of times that team i has won against team j. Simplifying further we get

$$\begin{aligned}
\ell(\theta; x) = \sum_{i,j} w_{ij} \log(P(i > j)) &= \sum_{i,j} w_{ij} \log\left(\frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j}}\right) \\
&= \sum_{i,j} w_{ij} \left[\log(e^{\lambda_i}) - \log(e^{\lambda_i} + e^{\lambda_j})\right] \\
&= \sum_{i,j} w_{ij} \left[\lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})\right].
\end{aligned}$$

Now to find the stationary point of our log likelihood we first need to find the partial derivatives of the above function. For $k \in \{1, ..., 20\}$, we have

$$\begin{aligned}\frac{\partial}{\partial \lambda_k}\ell(\theta; x) &= \sum_{i,j} w_{ij}\frac{\partial}{\partial \lambda_k}[\lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})] \\ &= \sum_{j} w_{kj}\frac{\partial}{\partial \lambda_k}[\lambda_k - \log(e^{\lambda_k} + e^{\lambda_j})] + w_{jk}\frac{\partial}{\partial \lambda_k}[\lambda_j - \log(e^{\lambda_k} + e^{\lambda_j})]\end{aligned}$$

since the partial derivative with respect to λ_k is only non-zero when λ_k is featured in the term being differentiated i.e. we only consider terms which involve team k. In particular,

$$\frac{\partial}{\partial \lambda_i}[\lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})] = 1 - \frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j}}$$

$$\frac{\partial}{\partial \lambda_j}[\lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})] = -\frac{e^{\lambda_j}}{e^{\lambda_i} + e^{\lambda_j}}$$

and all other partial derivatives will be zero. Hence we can write each partial derivative as

$$\begin{aligned}\frac{\partial}{\partial \lambda_k}\ell(\theta; x) &= \sum_{j} w_{kj}\left[1 - \frac{e^{\lambda_k}}{e^{\lambda_k} + e^{\lambda_j}}\right] + w_{jk}\left[-\frac{e^{\lambda_k}}{e^{\lambda_k} + e^{\lambda_j}}\right] \\ &= \sum_{j} w_{kj} - n_{kj}\left(\frac{e^{\lambda_k}}{e^{\lambda_k} + e^{\lambda_j}}\right)\end{aligned}$$

where $n_{kj} = w_{kj} + w_{jk}$ is the number of times team k and team j have played each other. Of course $n_{kk} = w_{kk} = 0$ since a team can't play against itself. We know that a stationary point is a point at which all partial derivatives are equal to zero, more specifically to find a stationary point (we will show later that it must be a maximum) we must solve the following system of equations:

$$\begin{aligned}\sum_{j} w_{1j} - n_{1j}\left(\frac{e^{\lambda_1}}{e^{\lambda_1} + e^{\lambda_j}}\right) &= 0 \\ \sum_{j} w_{2j} - n_{2j}\left(\frac{e^{\lambda_2}}{e^{\lambda_2} + e^{\lambda_j}}\right) &= 0 \\ &\vdots \\ \sum_{j} w_{20j} - n_{20j}\left(\frac{e^{\lambda_{20}}}{e^{\lambda_{20}} + e^{\lambda_j}}\right) &= 0.\end{aligned}$$

These equations are too complex to solve analytically so instead we will rely on numerical optimisation methods in R, using the `optim` function. By default `optim` will minimise the function you supply to it, however we are interested in finding the maximum of our log likelihood function. To resolve this we will use the negative log likelihood in our call to `optim` since minimising the negative log likelihood is equivalent to maximising the log likelihood function. Hence to find the MLE we will try to find the stationary point of

$$-\ell(\theta; x) = \sum_{i,j} w_{ij}[\log(e^{\lambda_i} + e^{\lambda_j}) - \lambda_i]$$

which is given by the solution to the system of equations

$$\begin{aligned}
\sum_j n_{1j}\left(\frac{e^{\lambda_1}}{e^{\lambda_1} + e^{\lambda_j}}\right) - w_{1j} &= 0 \\
\sum_j n_{2j}\left(\frac{e^{\lambda_2}}{e^{\lambda_2} + e^{\lambda_j}}\right) - w_{2j} &= 0 \\
&\vdots \\
\sum_j n_{20j}\left(\frac{e^{\lambda_{20}}}{e^{\lambda_{20}} + e^{\lambda_j}}\right) - w_{20j} &= 0.
\end{aligned}$$

Finding the point of the log likelihood where all partial derivatives are equal to zero will give us a stationary point, but it remains to check that this stationary point is a global maximum, ensuring that our estimate does indeed uniquely maximise the log likelihood. Firstly, we want to check that

$$\ell(\theta; x) = \sum_{i,j} w_{ij}[\lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})]$$

is a concave function. Noting that the sum of concave functions is also concave, it suffices to check that the term inside the sum

$$w_{ij}[\lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})]$$

is concave. Now notice that w_{ij} is a non-negative integer and so will not affect concavity of the term, so to check concavity of the log likelihood, we only need to check that

$$\lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})$$

is a concave function. This is nice since we aren't able to visualise the 20-dimensional log likelihood, but we can visualise the 3-dimensional plot given by $f(\lambda_i, \lambda_j) = \lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})$, shown in figure 1.

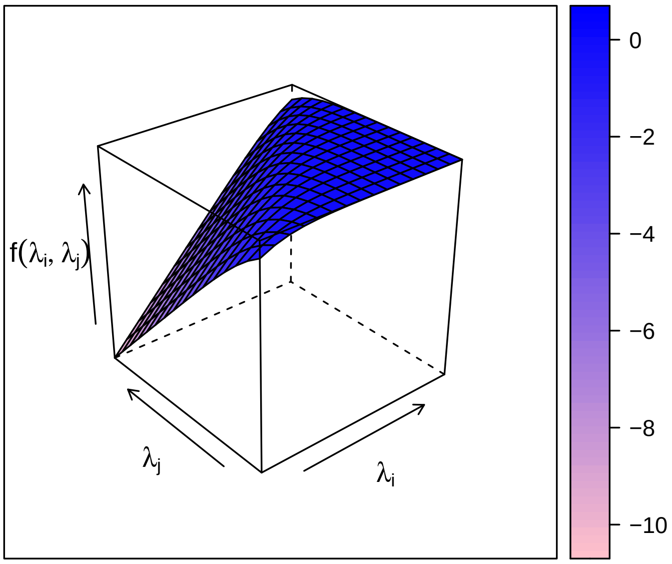


Figure 1: Plot of the function $f(\lambda_i, \lambda_j) = \lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})$

We can see that the surface does look concave. To show this, we know that λ_i is a linear function and it is known that linear functions are concave so the problem reduces again to checking that $-\log(e^{\lambda_i} + e^{\lambda_j})$ is concave, or equivalently that $\log(e^{\lambda_i} + e^{\lambda_j})$ is convex. We use the fact that the sum of log convex functions is itself log convex. Since $\log(e^{\lambda_i}) = \lambda_i$ is a linear function (and therefore also convex) it follows that e^{λ_i} and e^{λ_j} are both log convex so their sum is also log convex, that is, $\log(e^{\lambda_i} + e^{\lambda_j})$ is convex.

A more interesting proof of concavity is given by Hunter (2004) with an application of Hölder's inequality for sums which states that for positive numbers $c_1, .., c_N$ and $d_1, ..., d_N$ and $p \in (0, 1)$,

$$\sum_{k=1}^{N} c_k^p d_k^{1-p} \leq \left(\sum_{k=1}^{N} c_k\right)^p \left(\sum_{k=1}^{N} d_k\right)^{1-p}$$

with equality if and only if there exists some $\xi > 0$ such that $c_k = \xi d_k$ for all k. Taking logs of the above and multiplying by -1 we have

$$-\log\left(\sum_{k=1}^{N} c_k^p d_k^{1-p}\right) \geq -p \cdot \log\left(\sum_{k=1}^{N} c_k\right) - (1-p)\log\left(\sum_{k=1}^{N} d_k\right)$$

and we can use this to prove concavity of the function $f(\lambda_i, \lambda_j) = \lambda_i - \log(e^{\lambda_i} + e^{\lambda_j})$. For $p \in (0, 1)$ and any parameter vectors (α_i, α_j) and (β_i, β_j), we have

$$\begin{aligned}
&f(p\alpha_i + (1-p)\beta_i, p\alpha_j + (1-p)\beta_j) \\
&= p\alpha_i + (1-p)\beta_i - \log(e^{p\alpha_i+(1-p)\beta_i} + e^{p\alpha_j+(1-p)\beta_j}) \\
&= p\alpha_i + (1-p)\beta_i - \log[(e^{\alpha_i})^p(e^{\beta_i})^{1-p} + (e^{\alpha_j})^p(e^{\beta_j})^{1-p}] \\
&\geq p\alpha_i + (1-p)\beta_i - p\cdot\log(e^{\alpha_i} + e^{\alpha_j}) - (1-p)\log(e^{\beta_i} + e^{\beta_j}) \qquad (1) \\
&= p[\alpha_i - \log(e^{\alpha_i} + e^{\alpha_j})] + (1-p)[\beta_i - \log(e^{\beta_i} + e^{\beta_j})] \\
&= p\cdot f(\alpha_i, \alpha_j) + (1-p)f(\beta_i, \beta_j)
\end{aligned}$$

which is precisely the definition of concavity for the function f. As before, multiplying $f(\lambda_i, \lambda_j)$ by w_{ij} and summing over i and j won't affect concavity, hence the log likelihood is a concave function. The benefit of proving concavity using Hölder's inequality is that we can derive a sufficient condition that ensures the the concavity is strict. In particular, recall that we have equality in Hölder's inequality if and only if there exists some $\xi > 0$ such that $c_k = \xi d_k$ for all k. In our context we have equality if and only if there exists some $\xi > 0$ such that $e^{\alpha_i} = \xi e^{\beta_i}$ and $e^{\alpha_j} = \xi e^{\beta_j} \iff \alpha_i = \log(\xi) + \beta_i$ and $\alpha_j = \log(\xi) + \beta_j$. Combining the last two conditions yields $\alpha_i - \alpha_j = \beta_i - \beta_j$. Recall that the log likelihood is of the form

$$\ell(\theta; x) = \sum_{i,j} w_{ij} f(\lambda_i, \lambda_j).$$

Therefore for equality to hold in the concavity condition for the log likelihood we must have equality in the concavity condition for all functions of the form $f(\lambda_i, \lambda_j)$ which appear in the log likelihood. More specifically, we require $\alpha_i - \alpha_j = \beta_i - \beta_j$ for all i and j where $n_{ij} > 0$. Now consider the following assumption.

Assumption 1. *In every possible partition of the teams into two non-empty subsets, A and B, there is a team in A that is compared with a team in B at least once.*

Combining assumption 1 with the fact that $\alpha_i - \alpha_j = \beta_i - \beta_j$ for all i and j where $n_{ij} > 0$, we have $\alpha_i = \beta_i$ for all i. Note that for concavity of f to be strict we require that if equality holds in (1), then this implies that $(\alpha_i, \alpha_j) = (\beta_i, \beta_j)$. It then follows that assumption 1 ensures strict concavity of the log likelihood.

Assumption 2. *If in every possible partition of the teams into two non-empty subsets, A and B, there is a team in A that beats a team in B at least once.*

Additionally, Hunter (2004) shows that the log likelihood function is upper compact if assumption 2 is satisfied. Assumption 2 is stronger than assumption 1 and so under assumption 2 we know that the log likelihood is both upper compact and strictly concave. Informally, if the log likelihood function is upper compact then it must be bounded above and hence it cannot be monotonically increasing. Now notice that if we have a strictly concave function that is not monotonically increasing then the function must have a turning point, namely the global maximum. So we can conclude that under assumption 2, the MLE exists and it is unique.

Equivalently, we know that the negative log likelihood function is strictly convex and that it has a unique global minimum. We can use gradient descent methods in order to find this point, specifically we will use the L-BFGS-B method. This is one of many quasi-Newton methods, which is a class of methods used to find local extrema in models with a large number of parameters. Newton's method requires evaluating the inverse of the Hessian matrix at each iteration and for a model with N parameters the Hessian matrix has dimension NxN so finding the Hessian (and its inverse) becomes increasingly impractical. Quasi-Newton methods therefore use various procedures for approximating the Hessian in order to avoid finding its computationally complex, exact form and to speed up the algorithm. Performing the L-BFGS-B method on our negative log likelihood, we will obtain a local minimum. As long as assumption 2 holds, we know this point will be the unique global minimum.

As an example, let us consider all the outcomes from the 16/17 Premier League season. Since we aren't yet using a model which accounts for the outcome of a draw, we subset our data so as to remove any games ending in a draw and in this case we have 296 out of the 380 games remaining. Using this subsetted data we can construct the negative log likelihood generated by these 296 observations. We can do this by finding the log probability associated with each outcome, summing them up and multiplying by -1. Assume we have constructed this function in R, and called it `neg.log.lik`, we now make the following call to `optim`

```
optim(rep(-1,20), neg.log.lik, method = "L-BFGS-B")
```

which minimises the negative log likelihood using L-BFGS-B from an arbitrary starting value of -1 for each parameter. Finally, normalising the parameter estimates outputted by the above such that $\sum_{i=1}^{20} e^{\lambda_i} = 1$ yields the maximum likelihood estimates of the log strengths, λ_i, shown in table 4.

Remembering that ordering the teams with respect to their log strengths is equivalent to ordering them with respect to their strengths, we can rank the teams using these estimates. Ranking the teams from greatest log strength to lowest is precisely how we constructed the 'Model Rankings' column in table 1 from section 2.2, which is recreated to allow the reader to check the ordering (table 5).

Computing these maximum likelihood estimates as above takes R about 11 seconds to run which we want to speed up as later on we will want to repeat a similar calculation for every match week in order to continually update the model. Since there are 38 match weeks in a Premier League season this would be a very time consuming process. The L-BFGS-B method by default uses finite difference approximations in order to find the value of the gradient at each iteration. In addition, values of the gradient from

Team	log strength
Arsenal	-2.68
Bournemouth	-4.16
Middlesbrough	-5.32
Burnley	-4.33
Chelsea	-1.60
Crystal Palace	-4.34
Everton	-3.26
Hull	-4.76
Liverpool	-2.42
Leicester	-4.13
Man City	-2.23
Man United	-2.15
Sunderland	-5.41
Southampton	-4.09
Stoke	-4.21
Swansea	-4.43
Tottenham	-1.56
Watord	-4.29
West Brom	-3.99
West Ham	-4.08

Table 4: Maximum likelihood estimates of the log strengths, under the original Bradley-Terry model

Position	Model Rankings	True Rankings
1	Tottenham	Chelsea
2	Chelsea	Tottenham
3	Man United	Man City
4	Man City	Liverpool
5	Liverpool	Arsenal
6	Arsenal	Man united
7	Everton	Everton
8	West Brom	Southampton
9	West Ham	Bournemouth
10	Southampton	West Brom
11	Leicester	West Ham
12	Bournemouth	Leicester
13	Stoke	Stoke
14	Watford	Crystal Palace
15	Burnley	Swansea
16	Crystal Palace	Burnley
17	Swansea	Watford
18	Hull	Hull
19	Middlesbrough	Middlesbrough
20	Sunderland	Sunderland

Table 5: Rankings implied by the original Bradley-Terry model vs the true rankings, 16/17

previous iterations are used in the approximation of the Hessian matrix. Hence if we can supply the gradient to `optim` then it seems like the algorithm will be more efficient. Luckily for us we found the gradient analytically earlier in this section so we can construct a function for the gradient in R, call it `BTgrad`, and perform a similar computation to before by executing

```
optim(rep(-1,20), neg.log.lik, method = "L-BFGS-B", gr = BTgrad).
```

This does exactly the same as before but the gradient values can be calculated exactly rather than be approximated. Normalising the parameter estimates given by this code gives exactly the same values as before, but the run time has been reduced to around 1 second.

3.3 Inference for the model incorporating draws

Now we will look at applying the same ideas from the last section but instead we are considering the model which is also able to take draws into account. Remember that the extended model assigns probabilities in the following way

$$P(i > j) = \frac{\pi_i}{\pi_i + \pi_j + \nu\sqrt{\pi_i \pi_j}} = \frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i + \lambda_j}}}$$

$$P(i \leftrightarrow j) = \frac{\nu\sqrt{\pi_i \pi_j}}{\pi_i + \pi_j + \nu\sqrt{\pi_i \pi_j}} = \frac{\nu\sqrt{e^{\lambda_i + \lambda_j}}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i + \lambda_j}}}.$$

Also with $\theta = (\lambda_1, ..., \lambda_{20}, \nu) \in \mathbb{R}^{20} \times \mathbb{R}^+$, $x = (x_1, ..., x_n)$ the log likelihood function is given by

$$\ell(\theta; x) = \sum_{k=1}^{n} \log(P(x_k; \theta)) = \sum_{i,j} w_{ij} \log(P(i > j)) + \frac{d_{ij}}{2} \log(P(i \leftrightarrow j))$$

where w_{ij} is the number of times team i has beaten team j and d_{ij} is the number of times teams i and j have drawn. The fact that $d_{ij} = d_{ji}$ means that when we sum over all i and j we will count each draw between any teams twice, hence why the d_{ij} term is divided by 2. Substituting in the appropriate probabilities we have

$$\ell(\theta; x) = \sum_{i,j} w_{ij} \log\left(\frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i + \lambda_j}}}\right) + \frac{d_{ij}}{2} \log\left(\frac{\nu\sqrt{e^{\lambda_i + \lambda_j}}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i + \lambda_j}}}\right)$$

$$\ell(\theta;x) = \sum_{i,j} w_{ij}\left[\lambda_i - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right] + \frac{d_{ij}}{2}\left[\log(\nu) + \frac{\lambda_i+\lambda_j}{2} - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right].$$

Now we find the partial derivatives of the log likelihood, first with respect to the log strengths. For $k \in \{1, ..., 20\}$ we have

$$\frac{\partial}{\partial\lambda_k}\ell(\theta;x) = \frac{\partial}{\partial\lambda_k}\sum_{i,j} w_{ij}\left[\lambda_i - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right] + \frac{d_{ij}}{2}\left[\log(\nu) + \frac{\lambda_i+\lambda_j}{2} - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right]$$

$$\frac{\partial}{\partial\lambda_k}\ell(\theta;x) = \sum_{i,j} w_{ij}\frac{\partial}{\partial\lambda_k}\left[\lambda_i - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right] + \frac{d_{ij}}{2}\frac{\partial}{\partial\lambda_k}\left[\log(\nu) + \frac{\lambda_i+\lambda_j}{2} - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right].$$

Noticing that the partial derivatives with respect to λ_k in the above will be zero unless k is equal to either i or j, we can write the partial derivative with respect to λ_k of the log likelihood function as

$$\begin{aligned}\frac{\partial}{\partial\lambda_k}\ell(\theta;x) = \sum_{j} w_{kj}\frac{\partial}{\partial\lambda_k}\left[\lambda_k - \log\left(e^{\lambda_k} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_k+\lambda_j}}\right)\right] \\ + w_{jk}\frac{\partial}{\partial\lambda_k}\left[\lambda_j - \log\left(e^{\lambda_j} + e^{\lambda_k} + \nu\sqrt{e^{\lambda_j+\lambda_k}}\right)\right] \\ + d_{kj}\frac{\partial}{\partial\lambda_k}\left[\log(\nu) + \frac{\lambda_k+\lambda_j}{2} - \log\left(e^{\lambda_k} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_k+\lambda_j}}\right)\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial\lambda_k}\ell(\theta;x) = \sum_{j} w_{kj}\left[1 - \frac{e^{\lambda_k} + \frac{\nu}{2}\sqrt{e^{\lambda_k+\lambda_j}}}{e^{\lambda_k} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_k+\lambda_j}}}\right] \\ + w_{jk}\left[-\frac{e^{\lambda_k} + \frac{\nu}{2}\sqrt{e^{\lambda_k+\lambda_j}}}{e^{\lambda_k} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_k+\lambda_j}}}\right] + d_{kj}\left[\frac{1}{2} - \frac{e^{\lambda_k} + \frac{\nu}{2}\sqrt{e^{\lambda_k+\lambda_j}}}{e^{\lambda_k} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_k+\lambda_j}}}\right]\end{aligned}$$

$$\frac{\partial}{\partial\lambda_k}\ell(\theta;x) = \sum_{j} w_{kj} + \frac{d_{kj}}{2} - n_{kj}\left[\frac{e^{\lambda_k} + \frac{\nu}{2}\sqrt{e^{\lambda_k+\lambda_j}}}{e^{\lambda_k} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_k+\lambda_j}}}\right]$$

where $n_{kj} = w_{kj} + w_{jk} + d_{kj}$ is the number of times that teams k and j have played each other. We obtain, by similar reasoning, the partial derivative with respect to ν, we have

$$\begin{aligned}\frac{\partial}{\partial \nu}\ell(\theta;x) = \frac{\partial}{\partial \nu}\sum_{i,j} w_{ij}\left[\lambda_i - \log\left(e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right] \\ + \frac{d_{ij}}{2}\left[\log(\nu) + \frac{\lambda_i+\lambda_j}{2} - \log\left(e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial \nu}\ell(\theta;x) = \sum_{i,j} w_{ij}\frac{\partial}{\partial \nu}\left[\lambda_i - \log\left(e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right] \\ + \frac{d_{ij}}{2}\frac{\partial}{\partial \nu}\left[\log(\nu) + \frac{\lambda_i+\lambda_j}{2} - \log\left(e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial \nu}\ell(\theta;x) = \sum_{i,j} w_{ij}\left[-\frac{\sqrt{e^{\lambda_i+\lambda_j}}}{e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}}\right] \\ + \frac{d_{ij}}{2}\left[\frac{1}{\nu} - \frac{\sqrt{e^{\lambda_i+\lambda_j}}}{e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}}\right]\end{aligned}$$

$$\frac{\partial}{\partial \nu}\ell(\theta;x) = \sum_{i,j}\frac{d_{ij}}{2\nu} - \left(w_{ij}+\frac{d_{ij}}{2}\right)\left[\frac{\sqrt{e^{\lambda_i+\lambda_j}}}{e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}}\right].$$

To obtain the MLE we need to find the minimum of the negative log likelihood function, which is the solution to the following system of equations

$$\begin{aligned}\sum_j n_{1j}\left[\frac{e^{\lambda_1}+\frac{\nu}{2}\sqrt{e^{\lambda_1+\lambda_j}}}{e^{\lambda_1}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_1+\lambda_j}}}\right] - w_{1j} - \frac{d_{1j}}{2} &= 0 \\ \sum_j n_{2j}\left[\frac{e^{\lambda_2}+\frac{\nu}{2}\sqrt{e^{\lambda_2+\lambda_j}}}{e^{\lambda_2}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_2+\lambda_j}}}\right] - w_{2j} - \frac{d_{2j}}{2} &= 0 \\ &\vdots \\ \sum_j n_{20j}\left[\frac{e^{\lambda_{20}}+\frac{\nu}{2}\sqrt{e^{\lambda_{20}+\lambda_j}}}{e^{\lambda_{20}}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_{20}+\lambda_j}}}\right] - w_{20j} - \frac{d_{20j}}{2} &= 0 \\ \sum_{i,j}\left(w_{ij}+\frac{d_{ij}}{2}\right)\left[\frac{\sqrt{e^{\lambda_i+\lambda_j}}}{e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}}\right] - \frac{d_{ij}}{2\nu} &= 0.\end{aligned}$$

We need to check that the stationary point given by the solution to these equations is indeed a global minimum. As in the last section, we start by checking that the log likelihood function is concave. We want to show that

$$\ell(\theta; x) = \sum_{i,j} w_{ij} \left[\lambda_i - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right]$$
$$+ \frac{d_{ij}}{2}\left[\log(\nu) + \frac{\lambda_i+\lambda_j}{2} - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)\right]$$

is concave. Using the facts that the sum of concave functions is concave and that multiplying by a positive constant also preserves concavity, it suffices to show that the functions given by

$$\lambda_i - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right) \text{ and}$$
$$\log(\nu) + \frac{\lambda_i+\lambda_j}{2} - \log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)$$

are both concave. Also since linear functions are concave and it is known that the natural logarithm is strictly concave we only need to show that

$$-\log\left(e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}\right)$$

is concave. To get an idea of what the function looks like we set $\nu = 1$ in order to reduce the dimension, allowing us to visualise it. The function $f(x,y) = -\log\left(e^x + e^y + \sqrt{e^{x+y}}\right)$ shown in figure 2 does indeed look concave.

We can prove concavity of the function $g(\lambda_i, \lambda_j, \phi) = -\log\left(e^{\lambda_i} + e^{\lambda_j} + e^{\phi}\sqrt{e^{\lambda_i+\lambda_j}}\right)$ by using Hölder's inequality, noting that we have reparameterised $\nu > 0$ such that $\nu = e^{\phi}$. Recall the result we obtained from Hölder's inequality which states that, for positive numbers $c_1, ..., c_N$ and $d_1, ..., d_N$, and $p \in (0,1)$

$$-\log\left(\sum_{k=1}^{N} c_k^p d_k^{1-p}\right) \geq -p \cdot \log\left(\sum_{k=1}^{N} c_k\right) - (1-p)\log\left(\sum_{k=1}^{N} d_k\right)$$

with equality if and only if there exists some $\xi > 0$ such that $c_k = \xi d_k$ for all k. Now for $p \in (0,1)$ and any vectors of parameters $(\alpha_i, \alpha_j, \phi_1)$ and $(\beta_i, \beta_j, \phi_2)$ we have

$$g(p\alpha_i + (1-p)\beta_i, p\alpha_j + (1-p)\beta_j, p\phi_1 + (1-p)\phi_2)$$
$$= -\log\left(e^{p\alpha_i+(1-p)\beta_i} + e^{p\alpha_j+(1-p)\beta_j} + e^{p\phi_1+(1-p)\phi_2}\sqrt{e^{p\alpha_i+(1-p)\beta_i+p\alpha_j+(1-p)\beta_j}}\right)$$
$$= -\log\left(\left(e^{\alpha_i}\right)^p \left(e^{\beta_i}\right)^{1-p} + \left(e^{\alpha_j}\right)^p \left(e^{\beta_j}\right)^{1-p} + \left(e^{\phi_1}\right)^p \left(e^{\phi_2}\right)^{1-p} e^{\frac{(\alpha_i+\alpha_j)p+(\beta_i+\beta_j)(1-p)}{2}}\right)$$
$$= -\log\left(\left(e^{\alpha_i}\right)^p \left(e^{\beta_i}\right)^{1-p} + \left(e^{\alpha_j}\right)^p \left(e^{\beta_j}\right)^{1-p} + \left(e^{\phi_1+\frac{\alpha_i+\alpha_j}{2}}\right)^p \left(e^{\phi_2+\frac{\beta_i+\beta_j}{2}}\right)^{1-p}\right)$$
$$\geq p \cdot -\log\left(e^{\alpha_i} + e^{\alpha_j} + e^{\phi_1+\frac{\alpha_i+\alpha_j}{2}}\right) + (1-p) - \log\left(e^{\beta_i} + e^{\beta_j} + e^{\phi_2+\frac{\beta_i+\beta_j}{2}}\right) \tag{2}$$
$$= p \cdot g(\alpha_i, \alpha_j, \phi_1) + (1-p)g(\beta_i, \beta_j, \phi_2)$$

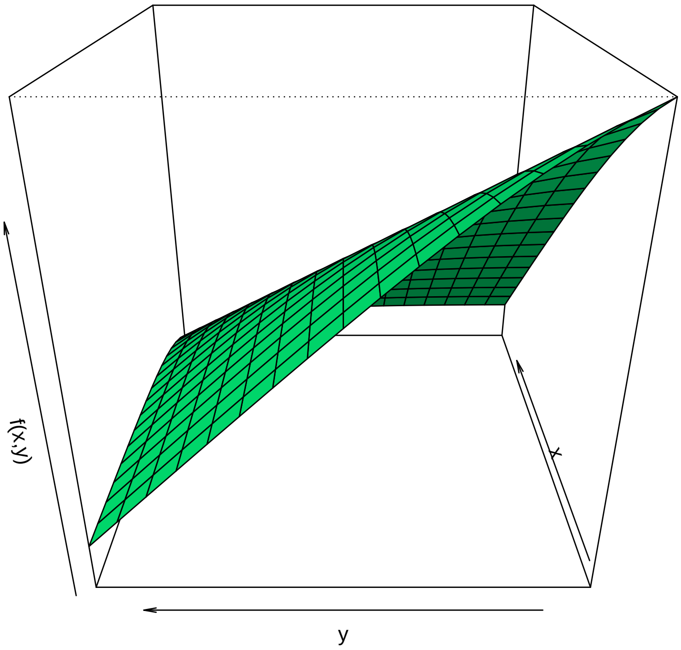


Figure 2: Plot of the function $f(x, y) = -\log\left(e^x + e^y + \sqrt{e^{x+y}}\right)$

which is exactly the condition required for concavity of the function g. We have equality in (2) if and only if there exists some $\xi > 0$ such that $e^{\alpha_i} = \xi e^{\beta_i}$, $e^{\alpha_j} = \xi e^{\beta_j}$ and $e^{\phi_1 + \frac{\alpha_i + \alpha_j}{2}} = \xi e^{\phi_2 + \frac{\beta_i + \beta_j}{2}}$. Taking logs changes the equations to $\alpha_i = \log(\xi) + \beta_i$, $\alpha_j = \log(\xi) + \beta_j$ and $\phi_1 + \frac{\alpha_i + \alpha_j}{2} = \log(\xi) + \phi_2 + \frac{\beta_i + \beta_j}{2}$. Notice that this function g appears in the log likelihood whenever $n_{ij} > 0$, so for equality to hold in the concavity condition for the full log likelihood, equality must hold for the concavity condition of the function g for all i and j such that $n_{ij} > 0$. More specifically, whenever $n_{ij} > 0$ we have

$$\alpha_i - \beta_i = \alpha_j - \beta_j = \phi_1 - \phi_2 + \frac{\alpha_i + \alpha_j - \beta_i - \beta_j}{2}$$

which implies that $\alpha_i - \beta_i = \alpha_j - \beta_j$ and $\phi_1 = \phi_2$. Now if assumption 1 is satisfied and there is at least one draw, then we have that $(\alpha_1, ..., \alpha_{20}, \phi_1) = (\beta_1, ..., \beta_{20}, \phi_2)$. Hence in this case the concavity is strict.

Again Hunter (2004) shows that the log likelihood function is upper compact if assumption 2 holds and there is at least one draw. Since assumption 2 is stronger than assumption 1, we know that if the log likelihood is upper compact then it is also strictly concave. Therefore under assumption 2 and the presence of at least one draw, we know that the MLE exists and is unique.

Now we can show how we obtained table 2 in section 2.2, where we found the strengths of all the teams at the end of the 16/17 Premier league season using this extended Bradley-Terry model. To accomplish this we use all 380 outcomes from that season and we sum all the log probabilities associated with each outcome given by our model to create the log likelihood function. Assume we have constructed the negative of this function and called it `neg.log.lik`, remembering that we are interested in the negative log likelihood since `optim` minimises functions by default. As we did in the previous section we use a quasi-Newton method to minimise our function by executing the code

```
optim(rep(1,21), neg.log.lik, method = "L-BFGS-B")
```

which initialises the minimisation arbitrarily from a value of 1 for all 21 parameters, where the first 20 parameters are the log strengths of the teams and the last parameter is ν which controls the prevalence of draws. Normalising the estimates of the log strengths given by the above code such that $\sum_{i=1}^{20} e^{\lambda_i} = 1$, we obtain the maximum likelihood estimates shown in table 6.

Team	log strength
Arsenal	-2.53
Bournemouth	-4.03
Middlesbrough	-4.95
Burnley	-4.44
Chelsea	-1.34
Crystal Palace	-4.44
Everton	-3.23
Hull	-4.77
Liverpool	-2.35
Leicester	-4.19
Man City	-2.25
Man United	-2.63
Sunderland	-5.43
Southampton	-4.03
Stoke	-4.11
Swansea	-4.44
Tottenham	-1.71
Watord	-4.44
West Brom	-4.11
West Ham	-4.11

Table 6: Maximum likelihood estimates of the log strengths, under the extended Bradley-Terry model

Ordering the teams from highest log strength to lowest is how we obtained the 'Model Rankings' column in table 2 from section 2.2, which we reproduce (table 7).

Position	Model Rankings	True Rankings
1	Chelsea	Chelsea
2	Tottenham	Tottenham
3	Man City	Man City
4	Liverpool	Liverpool
5	Arsenal	Arsenal
6	Man United	Man united
7	Everton	Everton
8	Bournemouth	Southampton
9	Southampton	Bournemouth
10	West Brom	West Brom
11	Stoke	West Ham
12	West Ham	Leicester
13	Leicester	Stoke
14	Swansea	Crystal Palace
15	Watford	Swansea
16	Burnley	Burnley
17	Crystal Palace	Watford
18	Hull	Hull
19	Middlesbrough	Middlesbrough
20	Sunderland	Sunderland

Table 7: Rankings implied by the extended Bradley-Terry model vs the true rankings, 16/17

The run time for this code is around 30 seconds which is far too impractical, so we use the same trick as before and construct a function which returns the value of the gradient for the negative log likelihood function, call it `BTDgrad`. Now we can execute the same code but we supply the gradient analytically to the `optim` function, so it doesn't need to approximate it. Executing

```
optim(rep(1,21), neg.log.lik, method = "L-BFGS-B", gr=BTDgrad)
```

provides us the same results yet only takes around 3 seconds, a significant improvement.

We know that since the model rankings are very similar to the true rankings that the maximum likelihood estimates for the log strengths are fairly sensible but it doesn't tell us much about how sensible our estimate for ν is. The estimate for this parameter was $\nu \approx 0.725$. To gauge whether this value makes sense we will consider the probability of two average teams drawing against each other, and we expect that this probability is somewhat close to the actual proportion of games that were drawn throughout the season. Let the log of the mean strength of all the teams equal a, then the probability of a draw between two 'average' teams is given by

$$\frac{\nu\sqrt{e^{\lambda_i+\lambda_j}}}{e^{\lambda_i}+e^{\lambda_j}+\nu\sqrt{e^{\lambda_i+\lambda_j}}} \approx \frac{0.725\sqrt{e^{a+a}}}{e^a+e^a+0.725\sqrt{e^{a+a}}} \approx \frac{0.725}{2+0.725} \approx 0.27.$$

In the 16/17 season 84 out of the 380 games ended in a draw, and $\frac{84}{380} \approx 0.22$. The two values are roughly the same, so we conclude that the value we have obtained for ν is fairly sensible.

3.4 Asymptotic distribution of the MLE

In this section we want to look into how we can quanitfy the uncertainty of our parameter estimates. In statistical inference it's important not only to estimate our parameters, but also to compute things such as the standard errors and confidence intervals of our estimates to allow us to know how much we can trust them. To derive standard errors for our estimates we will look into the asymptotic distibution of the MLE i.e. the distribution of the MLE as our sample size tends to infinity.

Suppose we have independent and identically distributed samples $x = (x_1, ..., x_n)$ from some distribution $f(\cdot;\theta)$, with θ the scalar parameter of interest, and the log likelihood is given by

$$\ell(\theta; x) = \sum_{i=1}^{n} \log(f(x_i;\theta)).$$

Then under what are commonly referred to as regularity conditions, which we take for granted here, we have asymptotic normality of the MLE, that is,

$$\sqrt{n}\left(\hat{\theta}-\theta_0\right) \text{ converges in distribution to } \mathcal{N}\left(0, \frac{1}{I(\theta_0)}\right)$$

where θ_0 is the true value of the parameter, $\hat{\theta}$ is the MLE and $I(\theta)$ is the Fisher information given by

$$I(\theta) = \mathbb{E}_\theta\left[\left(\frac{\partial}{\partial\theta}\ell(\theta;X)\right)^2\right],$$

with $X \sim f(\cdot;\theta_0)$.

Sketch of proof

We will cover the main ideas used to prove the above. Recall that $\hat{\theta}$ is the maximiser of $\ell(\theta;x)$ so we know that $\ell'(\hat{\theta};x) = 0$. Applying a first-order Taylor expansion to $\ell'(\hat{\theta};x)$ about $\hat{\theta} = \theta_0$ gives

$$\begin{aligned} &\ell'(\hat{\theta};x) \approx \ell'(\theta_0;x) + (\hat{\theta}-\theta_0)\ell''(\theta_0;x) \\ \iff\ & 0 \approx \sqrt{n}\ell'(\theta_0;x) + \sqrt{n}(\hat{\theta}-\theta_0)\ell''(\theta_0;x) \\ \iff\ & \sqrt{n}(\hat{\theta}-\theta_0) \approx -\sqrt{n}\frac{\ell'(\theta_0;x)}{\ell''(\theta_0;x)} = -\frac{\frac{1}{\sqrt{n}}\ell'(\theta_0;x)}{\frac{1}{n}\ell''(\theta_0;x)}. \end{aligned} \tag{3}$$

Note that we have set out to find the limiting distribution of $\sqrt{n}(\hat{\theta} - \theta_0)$ but we will now try to find the limiting distribution of its approximation, given in (3). First we want to find the limiting value of the denominator, $\frac{1}{n}\ell''(\theta_0; x)$. The weak law of large numbers tells us that

$$\begin{aligned}
\frac{1}{n}\ell''(\theta_0; x) = \frac{1}{n}\frac{\partial^2}{\partial\theta^2}\left[\sum_{i=1}^{n} \log f(x_i;\theta)\right]_{\theta=\theta_0} &= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2}{\partial\theta^2}\left[\log f(x_i;\theta)\right]_{\theta=\theta_0} \\
&\to \mathbb{E}_{\theta_0}\left[\frac{\partial^2}{\partial\theta^2}\left[\log f(X;\theta)\right]_{\theta=\theta_0}\right] \\
&= \mathbb{E}_{\theta_0}\left[\ell''(\theta_0; X)\right]
\end{aligned}$$

where the arrow denotes convergence in probability and $X \sim f(\cdot;\theta_0)$. We can show that this limiting value is equivalent to $-I(\theta_0)$. To do this note that

$$\ell'(\theta; X) = (\log f(X;\theta))' = \frac{f'(X;\theta)}{f(X;\theta)}$$

and by the quotient rule

$$\begin{aligned}
\ell''(\theta; X) = \left(\frac{f'(X;\theta)}{f(X;\theta)}\right)' &= \frac{f''(X;\theta)f(X;\theta) - f'(X;\theta)^2}{f(X;\theta)^2} \\
&= \frac{f''(X;\theta)}{f(X;\theta)} - \frac{f'(X;\theta)^2}{f(X;\theta)^2}.
\end{aligned}$$

Since $f(\cdot;\theta)$ is a probability density function we know that

$$\int f(x;\theta)dx = 1.$$

Differentiating both sides of this equation with respect to θ gives

$$\frac{\partial}{\partial\theta}\int f(x;\theta)dx = 0 \iff \int \frac{\partial}{\partial\theta} f(x;\theta)dx = 0 \iff \int f'(x;\theta)dx = 0 \quad (4)$$

where we have swapped the derivative and integral (allowed by the regularity conditions we have assumed). Taking derivatives with respect to θ again on both sides tells us that

$$\int f''(x;\theta)dx = 0.$$

Putting all of this together we have

$$\begin{aligned}
\mathbb{E}_{\theta_0}\left[\ell''(\theta_0;X)\right] &= \int \ell''(\theta_0;x)f(x;\theta_0)dx \\
&= \int \left(\frac{f''(x;\theta_0)}{f(x;\theta_0)} - \left(\frac{f'(x;\theta_0)}{f(x;\theta_0)}\right)^2\right) f(x;\theta_0)dx \\
&= \int f''(x;\theta_0)dx - \int \left(\frac{f'(x;\theta_0)}{f(x;\theta_0)}\right)^2 f(x;\theta_0)dx \\
&= 0 - \int \left(\frac{f'(x;\theta_0)}{f(x;\theta_0)}\right)^2 f(x;\theta_0)dx \\
&= -\mathbb{E}_{\theta_0}\left[\ell'(\theta_0;X)^2\right] = -I(\theta_0).
\end{aligned}$$

Hence we know that the denominator $\frac{1}{n}\ell''(\theta_0;x)$ converges in probability to $-I(\theta_0)$. Next we want to find the limiting distribution of the numerator of (3), $\frac{1}{\sqrt{n}}\ell'(\theta_0;x)$, and to help us do this we first derive the following auxiliary result:

$$\begin{aligned}
\mathbb{E}_\theta\left[\frac{\partial}{\partial\theta}\log f(X;\theta)\right] = \int \frac{\partial}{\partial\theta}\log(f(x;\theta))f(x;\theta)dx &= \int \frac{f'(x;\theta)}{f(x;\theta)}f(x;\theta)dx \\
&= \int f'(x;\theta)dx = 0 \quad (5)
\end{aligned}$$

where the last equality was shown in (4). Using this result we can write

$$\frac{1}{\sqrt{n}}\ell'(\theta_0;x) = \frac{1}{\sqrt{n}}\frac{\partial}{\partial\theta}\left[\sum_{i=1}^n \log f(x_i;\theta)\right]_{\theta=\theta_0} = \frac{1}{\sqrt{n}}\sum_{i=1}^n \frac{\partial}{\partial\theta}\left[\log f(x_i;\theta)\right]_{\theta=\theta_0}$$

where the term in the last sum has an expectation of zero as we showed in (5). This allows us to write

$$\begin{aligned}
&\frac{1}{\sqrt{n}}\sum_{i=1}^n \frac{\partial}{\partial\theta}\left[\log f(x_i;\theta)\right]_{\theta=\theta_0} \\
&= \sqrt{n}\left[\frac{1}{n}\sum_{i=1}^n \frac{\partial}{\partial\theta}[\log f(x_i;\theta)]_{\theta=\theta_0} - \mathbb{E}_{\theta_0}\left[\frac{\partial}{\partial\theta}[\log f(x_i;\theta)]_{\theta=\theta_0}\right]\right].
\end{aligned}$$

So the central limit theorem tells us that

$$\frac{1}{\sqrt{n}}\sum_{i=1}^n \frac{\partial}{\partial\theta}\left[\log f(x_i;\theta)\right]_{\theta=\theta_0} \text{ converges in distribution to } \mathcal{N}\left(0,\sigma^2\right)$$

with

$$\sigma^2 = \mathrm{Var}_{\theta_0}(\ell'(\theta_0;X)) = \mathbb{E}_{\theta_0}(\ell'(\theta_0;X)^2) - \left(\mathbb{E}_{\theta_0}(\ell'(\theta_0;X))\right)^2 = I(\theta_0) - 0^2$$

where the last equality comes from the definition of the Fisher information and the result shown in (5).

So far we have shown that $\frac{1}{n}\ell''(\theta_0; x)$ converges in probability to $-I(\theta_0)$ and $\frac{1}{\sqrt{n}}\ell'(\theta_0; x)$ converges in distribution to $\mathcal{N}(0, I(\theta_0))$. Now using a result known as Slutsky's theorem, we have that

$$-\frac{\frac{1}{\sqrt{n}}\ell'(\theta_0; x)}{\frac{1}{n}\ell''(\theta_0; x)} \text{ converges in distribution to } \frac{1}{I(\theta_0)}\mathcal{N}(0, I(\theta_0)) = \mathcal{N}\left(0, \frac{1}{I(\theta_0)}\right),$$

which gives our result. The significance of this result can be seen with the following inequality.

Cramér-Rao lower bound Let $x = (x_1, ..., x_n)$ be i.i.d samples from a probability density function $f(x; \theta_0)$. Then under some assumptions about regularity, for any unbiased estimator $T(x)$ of θ_0 we have

$$\operatorname{Var}(T(x)) \geq \frac{1}{nI(\theta_0)}.$$

Since the MLE is unbiased, looking at the asymptotic distribution of the MLE we can show that it achieves this lower bound. Noticing that if we do have $\sqrt{n}\left(\hat{\theta} - \theta_0\right) \sim \mathcal{N}\left(0, \frac{1}{I(\theta_0)}\right)$, then equivalently we have

$$\hat{\theta} - \theta_0 \sim \mathcal{N}\left(0, \frac{1}{nI(\theta_0)}\right) \iff \hat{\theta} \sim \mathcal{N}\left(\theta_0, \frac{1}{nI(\theta_0)}\right).$$

It can be seen that the asymptotic variance of the MLE achieves the Cramér-Rao lower bound. This means that if we have a large number of samples then we can be confident that out of any possible unbiased estimator, the MLE will have the smallest variance, justifying its popularity.

So far we have only considered the case where θ is a single parameter, however the asymptotic result we derived extends easily to the case where θ is a vector of parameters. In this case, $\hat{\theta}$ is asymptotically multivariate normal with mean vector θ_0 and covariance matrix $\frac{1}{n}\mathcal{I}(\theta_0)^{-1}$ where $\mathcal{I}(\theta_0)$ is the Fisher information matrix which satisfies

$$[\mathcal{I}(\theta)]_{i,j} = \mathbb{E}_\theta\left[\left(\frac{\partial}{\partial\theta_i}\ell(\theta; X)\right)\left(\frac{\partial}{\partial\theta_j}\ell(\theta; X)\right)\right].$$

3.5 Quantifying uncertainty

We can use the asymptotic distribution to allow us to get an idea of the standard errors of our parameter estimates. Recall that in the sample limit the covariance matrix of the MLE is given by $\frac{1}{n}\mathcal{I}(\theta_0)^{-1}$. Of course we don't know the value of θ_0 so we can't compute the actual covariance matrix of the MLE. However we can approximate the covariance matrix of $\hat{\theta}$ with the inverse of the observed Fisher information, where the observed Fisher information is given by

$$\hat{\mathcal{I}} = -\frac{\partial}{\partial\theta\partial\theta^T}\left[\ell(\theta; x)\right]_{\theta=\hat{\theta}}.$$

This is nothing more than the negative of the matrix of second derivatives of the log likelihood, evaluated at the MLE. The matrix of second derivatives (Hessian) is fairly easy to obtain, which we multiply by -1 and then invert the resulting matrix to obtain an approximate covariance matrix. We can then take the square roots of the diagonal entries of this covariance matrix to obtain the standard errors for each parameter estimate.
Recall that we used the function `optim` to find the MLEs. We can simply add the argument "Hessian=TRUE" in our call to `optim` and the function will return the values of the Hessian evaluated at the MLE, which can then be used to derive the standard errors. Unfortunately in our current setup it isn't that simple. Remember that our MLE is the maximum of the log likelihood subject to the constraint

$$\sum_{i=1}^{20} e^{\lambda_i} = 1,$$

and we obtained our MLE's by minimising the unconstrained negative log likelihood and then normalising the parameters. This gives the correct MLEs however it will not return the correct Hessian since `optim` will return the Hessian of the unconstrained negative log likelihood evaluated at the unnormalised parameter estimates. Whereas we are interested in obtaining the Hessian of the constrained negative log likelihood evaluated at the normalised parameter estimates. We can solve this by considering an equivalent way of enforcing our constraint which is to fix the log strength of one of the teams. We require the constraint because in our model we can't distinguish between the parameter vectors $(\lambda_1, ..., \lambda_{20)}$ and $(\lambda_1 + c, ..., \lambda_{20} + c)$ hence if we fix the value of λ_1 then we essentially fix the value of c, making the parameters identifiable.

We want to find the standard errors associated with our parameter estimates from section 3.3, shown in table 8.

In the interest of getting the same parameter estimates as well as the standard errors we reconstruct the negative log likelihood function but we fix $\lambda_1 = -2.53$. We then minimise the resulting function of 20 parameters and since we have removed the parameter λ_1 our function is now unconstrained and `optim` will return the appropriate value of the Hessian. Note that we want the negative Hessian of our log likelihood which is the same as the Hessian of the negative log likelihood, so all that remains is to find the square roots of the diagonals of the inverse of the Hessian given by `optim` to obtain the standard errors. The results are shown in table 9.

These estimates are based off of the 380 results from the 16/17 season, hence we are assuming that a sample size of 380 is sufficiently large so that the distribution of the MLE is well approximated by its asymptotic distribution. An alternative method of calculating the standard errors is what is known as bootstrapping.

Imagine that instead of having only 1 sample of size 380, we had n samples of size 380. Then we could calculate the MLE for all n samples and the standard deviations of the n parameter estimates would give us the standard

Parameter	MLE
λ_1 (Arsenal)	-2.53
λ_2 (Bournemouth)	-4.03
λ_3 (Middlesbrough)	-4.95
λ_4 (Burnley)	-4.44
λ_5 (Chelsea)	-1.34
λ_6 (Crystal Palace)	-4.44
λ_7 (Everton)	-3.23
λ_8 (Hull)	-4.77
λ_9 (Liverpool)	-2.35
λ_{10} (Leicester)	-4.19
λ_{11} (Man City)	-2.25
λ_{12} (Man United)	-2.63
λ_{13} (Sunderland)	-5.43
λ_{14} (Southampton)	-4.03
λ_{15} (Stoke)	-4.11
λ_{16} (Swansea)	-4.44
λ_{17} (Tottenham)	-1.71
λ_{18} (Watord)	-4.44
λ_{19} (West Brom)	-4.11
λ_{20} (West Ham)	-4.11
ν	0.73

Table 8: Maximum likelihood estimates for all parameters in the extended Bradley-Terry model, 16/17

Parameter	MLE	Standard error
λ_1 (Arsenal)	-2.53	-
λ_2 (Bournemouth)	-4.03	0.60
λ_3 (Middlesbrough)	-4.95	0.62
λ_4 (Burnley)	-4.44	0.61
λ_5 (Chelsea)	-1.34	0.68
λ_6 (Crystal Palace)	-4.44	0.61
λ_7 (Everton)	-3.23	0.59
λ_8 (Hull)	-4.77	0.62
λ_9 (Liverpool)	-2.35	0.62
λ_{10} (Leicester)	-4.19	0.60
λ_{11} (Man City)	-2.25	0.62
λ_{12} (Man United)	-2.63	0.61
λ_{13} (Sunderland)	-5.43	0.65
λ_{14} (Southampton)	-4.03	0.60
λ_{15} (Stoke)	-4.11	0.60
λ_{16} (Swansea)	-4.44	0.61
λ_{17} (Tottenham)	-1.71	0.65
λ_{18} (Watord)	-4.44	0.61
λ_{19} (West Brom)	-4.11	0.60
λ_{20} (West Ham)	-4.11	0.60
ν	0.73	0.10

Table 9: Standard errors implied by the approximate asymptotic distribution

errors. Unfortunately the 16/17 Premier League season was only played once so we have access to one sample. To get around this we can resample with replacement from our data 380 times to create a new sample. We carry out the following procedure $n = 10^5$ times:

1. Resample from the data with replacement 380 times.
2. Compute the MLEs for this new sample.
3. Store this vector of MLEs.

After performing this procedure we will have 10^5 estimates for each parameter, allowing us to compute the standard deviation for each (table 10).

Parameter	MLE	Standard error (Asymptotic)	Standard Error (Bootstrap)
λ_1 (Arsenal)	-2.53	-	0.75
λ_2 (Bournemouth)	-4.03	0.60	0.76
λ_3 (Middlesbrough)	-4.95	0.62	0.74
λ_4 (Burnley)	-4.44	0.61	0.79
λ_5 (Chelsea)	-1.34	0.68	0.47
λ_6 (Crystal Palace)	-4.44	0.61	0.82
λ_7 (Everton)	-3.23	0.59	0.74
λ_8 (Hull)	-4.77	0.62	0.80
λ_9 (Liverpool)	-2.35	0.62	0.79
λ_{10} (Leicester)	-4.19	0.60	0.78
λ_{11} (Man City)	-2.25	0.62	0.69
λ_{12} (Man United)	-2.63	0.61	0.72
λ_{13} (Sunderland)	-5.43	0.65	0.83
λ_{14} (Southampton)	-4.03	0.60	0.74
λ_{15} (Stoke)	-4.11	0.60	0.71
λ_{16} (Swansea)	-4.44	0.61	0.80
λ_{17} (Tottenham)	-1.71	0.65	0.72
λ_{18} (Watord)	-4.44	0.61	0.78
λ_{19} (West Brom)	-4.11	0.60	0.75
λ_{20} (West Ham)	-4.11	0.60	0.75
ν	0.73	0.10	0.10

Table 10: Standard errors given by asymptotic distribution vs bootstrap standard errors

We can see that both methods yield similar results. Better yet we can plot the distributions of the MLEs given by both methods. We arbitrarily show the densities associated with parameters λ_2, λ_9 and ν (figures 3, 4 and 5).

It seems that the asymptotic and bootstrapping distributions are fairly similar. The small differences in the distributions can likely be accounted for by the fact that the two methods aren't strictly comparable. This is because in the bootstrapping method we calculate the MLEs by minimising the negative log likelihood with respect to the constraint

$$\sum_{i=1}^{20} e^{\lambda_i} = 1,$$

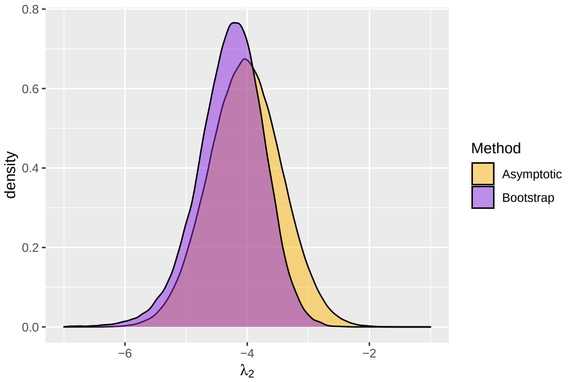


Figure 3: Distribution of λ_2

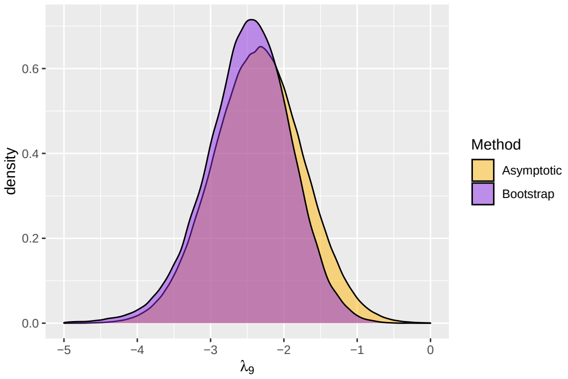


Figure 4: Distribution of λ_9

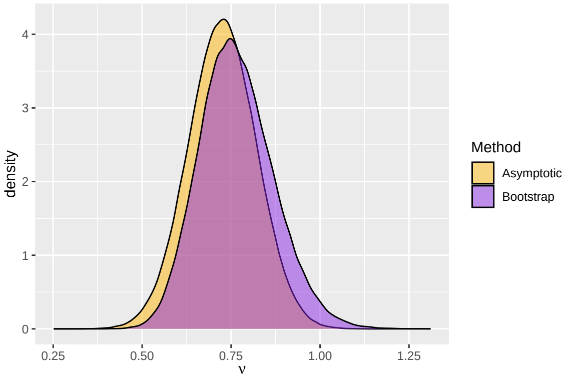


Figure 5: Distribution of ν

whereas to make finding the asymptotic distribution possible we had to remove one of the parameters and then minimise the resulting negative log likelihood. Hence the two methods are computing slightly different things. The plots are however very similar, potentially justifying our asymptotic assumptions.

3.6 MLE on simulated data

We have provided justification for why our model may be appropriate in this context. In this section we will simulate a season of games using our model, and then we will find the MLE of this simulated data. This will allow us to gauge how effective our estimation method is without having to worry about whether the model is an accurate reflection of how the data is really generated. We begin by randomly creating the 'true' values of the parameters, which we can then use to calculate outcome probabilities for all games in our hypothetical season. Using the inverse transform method we can simulate how the outcomes of all the games might happen giving us a simulated season of football games.

To understand how the inverse transform method works we give an example. Suppose the probability of team A beating team B is 0.5, the probability of a draw is 0.3 and the probability that team B beats team A is 0.2. Now to simulate the outcome of a game between A and B, we first generate $u \sim \text{uniform}[0, 1]$. Then the outcome is a win for A if $0 < u < 0.5$, a draw if $0.5 < u < 0.8$ and a win for B if $0.8 < u < 1$. Since u is uniform it is straightforward to see that this procedure will give each of the outcomes with the correct probability.

After simulating a season of matches, we compute the MLEs of our simulated data to try and recover the true values of the parameters (table 11). We can see that most of the estimates are close to the true values, so it seems that one season of data is sufficient in reliably estimating the parameters. Here we had one season worth of simulated data, which is a sample size of 380 (each team plays every other team twice $20 \times 19 \times 2 = 380$), and we obtained a mean squared error of 0.001. Let us look at how different sample sizes affect the mean squared error (figure 6).

We know that the MLE is a consistent estimator meaning that it converges to the true values of the parameters as the sample size tends to infinity. Hence we would expect the mean squared error of the MLE to tend to zero as the sample size increases, which figure 6 confirms.

Parameter	True value	MLE
π_1 (Arsenal)	0.027	0.027
π_2 (Bournemouth)	0.003	0.004
π_3 (Middlesbrough)	0.025	0.039
π_4 (Burnley)	0.026	0.052
π_5 (Chelsea)	0.054	0.022
π_6 (Crystal Palace)	0.016	0.036
π_7 (Everton)	0.048	0.047
π_8 (Hull)	0.045	0.127
π_9 (Liverpool)	0.082	0.047
π_{10} (Leicester)	0.121	0.157
π_{11} (Man City)	0.014	0.012
π_{12} (Man United)	0.001	0.001
π_{13} (Sunderland)	0.108	0.103
π_{14} (Southampton)	0.037	0.032
π_{15} (Stoke)	0.060	0.052
π_{16} (Swansea)	0.061	0.052
π_{17} (Tottenham)	0.049	0.036
π_{18} (Watord)	0.119	0.063
π_{19} (West Brom)	0.044	0.039
π_{20} (West Ham)	0.060	0.052
ν	1.45	1.52

Table 11: MLEs given by the simulated data vs the parameters' true values

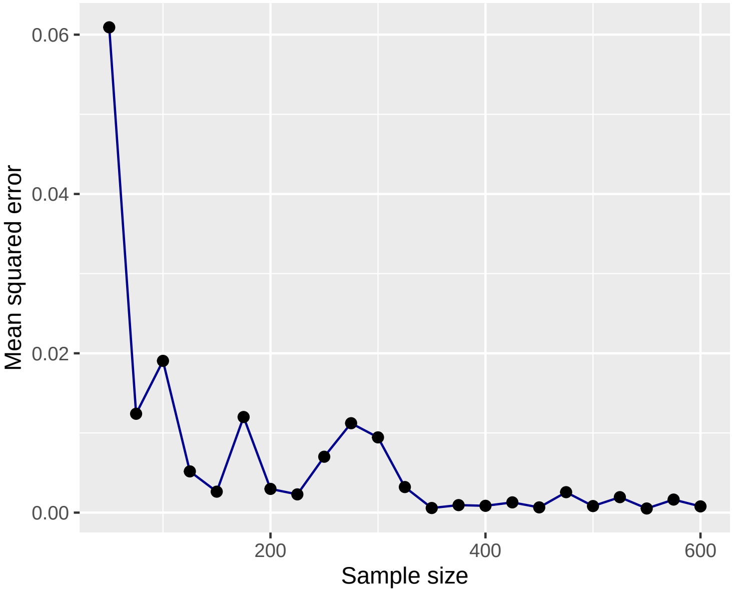


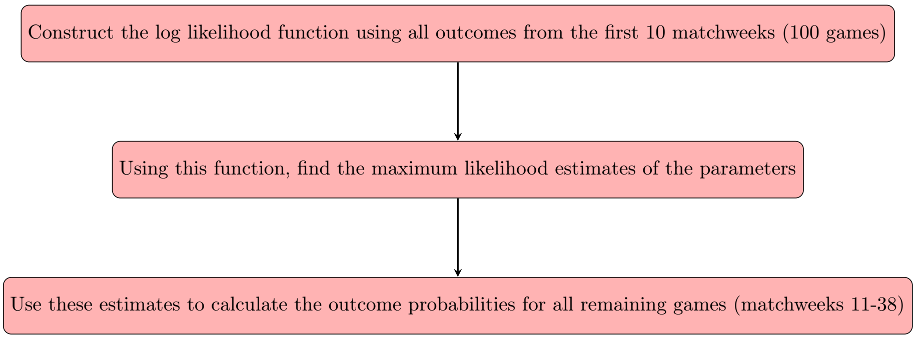
Figure 6: Consistency of the MLE

3.7 Predictions

3.7.1 Static model

Now that we have a suitable model and a method for inferring the parameters of this model we can begin to consider how we can make predictions. In this paper we will use the data from the 16/17 Premier League season to perform any data analysis or to determine which of our models is best, and we then test our model on a new season. In particular we will try to predict all games from matchweek 11 onward, from the 16/17 season. Recall from the previous section that there are some conditions we require to ensure that the MLE exists, hence to ensure such conditions are met we only make predictions if we have data from at least 10 previous matchweeks i.e. matchweek 11 is the first set of games we try to predict.

As a baseline, we first see how a 'static' model performs. We call the model static because we don't allow the team strengths to vary through time. The procedure goes as follows:



Construct the log likelihood function using all outcomes from the first 10 matchweeks (100 games)

Using this function, find the maximum likelihood estimates of the parameters

Use these estimates to calculate the outcome probabilities for all remaining games (matchweeks 11-38)

Carrying out the above procedure in R, we report our first 15 attempted predictions to get an idea of the results. Below is a table with the outcome probabilities given by the model alongside the actual outcomes of the games.

P(home win)	P(draw)	P(away win)	Home team	Away team	Score (Home-Away)
0.81	0.16	0.03	Bournemouth	Sunderland	1-2
0.49	0.33	0.18	Burnley	Crystal Palace	3-2
0.61	0.28	0.11	Chelsea	Everton	5-0
0.78	0.18	0.04	Man City	Middlesbrough	1-1
0.25	0.35	0.40	West Ham	Stoke	1-1
0.43	0.34	0.23	Arsenal	Tottenham	1-1
0.11	0.28	0.60	Hull	Southampton	2-1
0.52	0.32	0.16	Leicester	West Brom	1-2
0.73	0.22	0.06	Liverpool	Watford	6-1
0.05	0.21	0.73	Swansea	Man United	1-3
0.04	0.18	0.78	Crystal Palace	Man City	1-2
0.70	0.23	0.07	Everton	Swansea	1-1
0.11	0.28	0.61	Man United	Arsenal	1-1
0.05	0.21	0.74	Southampton	Liverpool	0-0
0.25	0.35	0.40	Stoke	Bournemouth	0-1

Table 12: Forecasted probabilities vs actual outcomes

Note that if the probabilities don't sum to 1 it's only because the probabilities have been rounded to 2 decimal places, and it is always the case that the exact probabilities sum to 1 as we would expect.

Now that we have made our first set of predictions, we need some methods to assess how good these predictions actually are. The most important measure of a model's accuracy is of course how many of the games it correctly predicts. Since the model outputs probabilities we will take the biggest of the three probabilities as the predicted outcome. For example, for the last result in table 12, 0.40 is the biggest probability so in this case the model has predicted the outcome to be an away win. Then looking at the actual result we see that the game finished 1-0 to the away team, hence our model has correctly classified this game. Checking all of the games we have attempted to predict we find that the model correctly classifies 155 out of the 280 games, giving a classification accuracy of around 55.4%. To get an idea of how good or bad this is we need to find the null classification, which is just the percentage of the games that ended in the most frequent outcome. As in most sports the home team has the advantage and so the most frequent outcome is home win. In the games we are predicting, roughly 51.8% of them ended in a home win meaning that the model does perform better than a classifier which simply predicts that every game will be a home win.

Another tool we will use is what is known as a confusion table which can give us information about where our classifier does well and where it doesn't. The confusion table for these predictions is given by

		Prediction		
		Home	Draw	Away
Actual	Home win	94	7	44
	Draw	26	2	30
	Away win	16	2	59

Table 13: Confusion table for static predictions

To confirm that the table is valid we can see that the sum of the diagonals of the table is 155, the same as the number of correctly classified games, and the sum of all the numbers is 280, the total number of games we tried to predict. The most notable information we get from this table is that our model hasn't predicted the outcome of a draw many times, meaning if we could somehow incorporate a better way of predicting draws we could improve the classifier. On the other hand, draws are somewhat unpredictable so it is potentially a good thing that our model rarely predicts a draw to be the most probable outcome.

The final measure we will use to assess the quality of our predictions is the rank probability score (RPS) which is defined as

$$RPS = \frac{1}{r-1}\sum_{i=1}^{r}\left(\sum_{j=1}^{i}(p_j - e_j)\right)^2$$

where r is the number of possible outcomes, p_j is the predicted probability of outcome j and e_j is the actual probability of outcome j i.e. e_j is 1 if j was the observed outcome and 0 if not. In our case the number of outcomes is $r = 3$ and we define outcome 1 to be a home win, outcome 2 to be a draw and outcome 3 to be an away win. The RPS is between 0 and 1 with lower scores corresponding to better predictions. The RPS is a strictly proper scoring rule, which means that it is uniquely minimised by the true probabilities. To see this notice how the best prediction is the one which predicts with certainty the observed outcome, in which case p_j is equal to e_j for all j and so the RPS will attain its optimal value of 0.

There are many other strictly proper scoring rules, but we use the RPS because it is sensitive to distance, arguably making it the most appropriate probabilistic scoring rule in the context of football (Constantinou and Fenton, 2012). To demonstrate what we mean by this imagine that we have predicted the outcome of a game to be an away win, then if the game actually ends in a draw then our prediction seems to be better than if the outcome was a home win. A draw seems 'closer' to an away win than a home win does, and we want the score to reflect this. To show that the RPS has this property consider we are in the first scenario where we predict an away win (with certainty) and the actual outcome is a draw, we have $(p_1, p_2, p_3) = (0, 0, 1)$ and $(e_1, e_2, e_3) = (0, 1, 0)$. The RPS for this prediction is given by

$$RPS = \frac{1}{2}\sum_{i=1}^{3}\left(\sum_{j=1}^{i}(p_j - e_j)\right)^2 = \frac{1}{2}[(p_1 - e_1)^2 + ((p_1 - e_1) + (p_2 - e_2))^2 + ((p_1 - e_1) + (p_2 - e_2) + (p_3 - e_3))^2]$$

$$RPS = \frac{1}{2}\left[0^2 + (0-1)^2 + (0-1+1)^2\right] = \frac{1}{2}.$$

For comparison now imagine we are in the scenario where we still predict an away win but the actual outcome is a home win, we have $(p_1, p_2, p_3) = (0, 0, 1)$ and $(e_1, e_2, e_3) = (1, 0, 0)$. The RPS in this case is

$$RPS = \frac{1}{2}[(p_1 - e_1)^2 + ((p_1 - e_1) + (p_2 - e_2))^2 + ((p_1 - e_1) + (p_2 - e_2) + (p_3 - e_3))^2]$$

$$RPS = \frac{1}{2}\left[(-1)^2 + (-1+0)^2 + (-1+0+1)^2\right] = 1$$

The first scenario gives a lower RPS so would be considered as the better prediction, as we wanted. For these reasons the RPS seems like a reasonable measure of success for probabilistic predictions of football games.

To demonstrate how the RPS works with our actual predictions we show the RPS associated with our first 15 predictions in table 14.

P(home win)	P(draw)	P(away win)	Home team - Away team	Score (Home-Away)	RPS
0.81	0.16	0.03	Bournemouth - Sunderland	1-2	0.799
0.49	0.33	0.18	Burnley - Crystal Palace	3-2	0.148
0.61	0.28	0.11	Chelsea - Everton	5-0	0.084
0.78	0.18	0.04	Man City - Middlesbrough	1-1	0.309
0.25	0.35	0.40	West Ham - Stoke	1-1	0.113
0.43	0.34	0.23	Arsenal - Tottenham	1-1	0.119
0.11	0.28	0.60	Hull - Southampton	2-1	0.577
0.52	0.32	0.16	Leicester - West Brom	1-2	0.488
0.73	0.22	0.06	Liverpool - Watford	6-1	0.038
0.05	0.21	0.73	Swansea - Man United	1-3	0.038
0.04	0.18	0.78	Crystal Palace - Man City	1-2	0.024
0.70	0.23	0.07	Everton - Swansea	1-1	0.248
0.11	0.28	0.61	Man United - Arsenal	1-1	0.191
0.05	0.21	0.74	Southampton - Liverpool	0-0	0.273
0.25	0.35	0.40	Stoke - Bournemouth	0-1	0.213

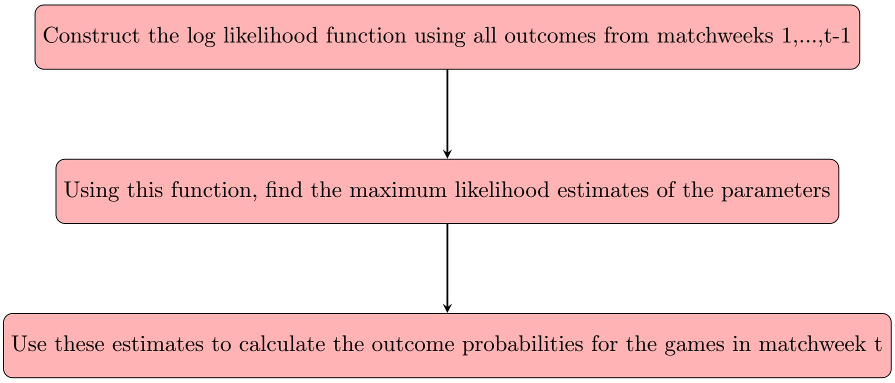
Table 14: RPS for various predictions

We can see that the RPS is almost optimal for the row highlighted green since the correct result was predicted with high probability. Also the red and orange rows demonstrate again why the RPS is appropriate in our context. Notice how both of these rows predict a home win with comparable levels of confidence. Although both predictions were wrong, the game in orange ended in a draw and so it has a lower (better) RPS than the game in red which ended in an away win.

Since we have an RPS associated with each of our predictions, we will summarise this information by only reporting the mean of all of the rank probability scores. For all 280 of our predictions the mean RPS we obtain is 0.211. Given the abstract definition of the RPS, knowing the mean RPS for our predictions is 0.211 doesn't really mean a lot to us. It will however be useful for us to compare the relative performance of our different models.

3.7.2 Dynamic model

Although the static model served as a simple introduction to how we will predict match outcomes, the method seemed like it wasn't using all of the data. To solve this we will implement a more 'dynamic' model where we recompute the estimates of our parameters every matchweek, using the data available from all previous games. For $t = 11, ..., 38$ we carry out the following procedure.


Construct the log likelihood function using all outcomes from matchweeks 1,...,t-1
Using this function, find the maximum likelihood estimates of the parameters
Use these estimates to calculate the outcome probabilities for the games in matchweek t

The predictions for matchweek 11 will be the same as in the static model however all subsequent predictions will differ. Evaluating the predictions given by this dynamic model we have a classification accuracy of 53.9%, which is slightly worse than the static model. The mean RPS is 0.207 which is a small improvement.

The fact that the most important measure, the percentage of games correctly classified, has decreased is a bit concerning. This could be because our model doesn't take into account how long ago a game was played. So although our model is now making full use of the data, it is potentially too sensitive to games that were played a long time ago.

3.7.3 Time-weighted model

Intuitively, it seems that when we are trying to predict the relative strengths of all the teams we should place more emphasis on the games that were played more recently. For example, a team may have a poor start to the season but they may start playing better later on. Since the team starts playing well and wins many of their games it should be the case that we predict their strength to be relatively high. However up until now we have been weighting all matches equally so our estimate of the team's strength will be based equally on their good recent performances and their worse older performances meaning we may underestimate the current strength of the team. In order to allow us to use all of the data we have available but not risk putting too much emphasis on older games we will investigate how well a time-weighted model performs. To incorporate a time component into the model we multiply each term in the log likelihood function by some function $\phi(t)$, where t is the time since the match in question was played. Recall that the log likelihood function is of the form

$$\ell(\theta; x) = \sum_{i,j} w_{ij} \log(P(i > j)) + \frac{d_{ij}}{2} \log(P(i \leftrightarrow j))$$

and we want to weight each term in the above sum such that more recent games are weighted more heavily. Hence we can propose a time-weighted model given by

$$\ell(\theta; x) = \sum_{i,j} \phi(t_{ij}) \left\{ w_{ij} \log(P(i > j)) + \frac{d_{ij}}{2} \log(P(i \leftrightarrow j)) \right\}$$

where t_{ij} represents the number of weeks since the match between teams i and j was played. Note that this equation only makes sense if no teams have played each other more than once, since if i and j have played multiple games then there are multiple values for t_{ij}. In practice, we multiply the log probabilities associated with each of the outcomes with $\phi(t)$, where t is the number of weeks ago the outcome ocurred. We then sum all of these terms to obtain the log likelihood.

Remember that we want less weighting on games further in the past, hence we should select $\phi(t)$ to be some decreasing function. The first option we shall consider is

$$\phi(t_{ij}) = \begin{cases} 1, & \text{if } t_{ij} \leq t_0. \\ 0, & \text{otherwise.} \end{cases}$$

which has the effect of only taking into account games that were played within the last t_0 weeks and disregarding any older results. We visualise the weighting function for $t_0 = 20$ in figure 7.

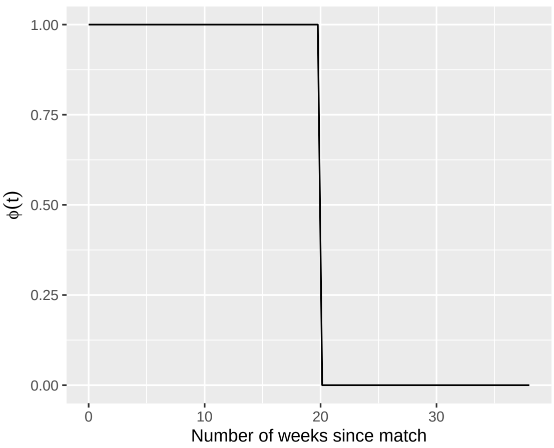


Figure 7: Weighting function with $t_0 = 20$

The log likelihood function for this model takes a similar form as the non-weighted model, it just contains fewer terms. Therefore, the conditions we derived in section 3.3 for the MLEs to exist are the same in this case. The only difference is that since we are restricting the number of games we use, we make it less likely that those conditons will be met.

We will find the optimal value for t_0 and compare it to our non-weighted model. For values of t_0 that are less than or equal to 3, R fails to output an estimate of the parameters. This is most likely because with only a small number of games being considered, the conditions we found in section 3.3, which guarantee a unique maximum to exist, have not been met. However, for $t_0 \geq 4$ we have no problems and figure 8 shows how the value of t_0 affects the classification accuracy.

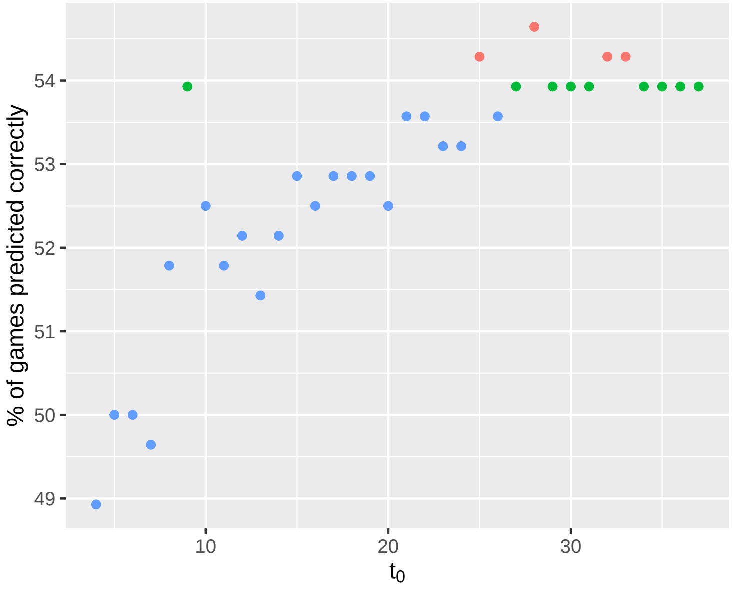


Figure 8: How t_0 affects classification accuracy

Since there are 38 matchweeks in a season, the maximum number of weeks we take into account is when we are trying to predict the last matchweek, in which case we have 37 previous weeks of data. Hence any value of t_0 greater than or equal to 37 means we use all previous games, the same as the dynamic model with no weighting. In the plot, the green points are values of t_0 which attain the same accuracy as the dynamic model with no weighting, whereas the blue and red points respectively correspond to worse and better accuracy. As shown in the plot we achieve a maximum accuracy of 54.6% for $t_0 = 28$. Although it is clear that generally for larger values of t_0 we attain higher accuracy, the accuracy appears to change somewhat sporadically and so it would be unwise to conclude that there is something special happening at $t_0 = 28$.

Another option for our weighting function is given by

$$\phi(t_{ij}) = e^{-wt_{ij}}$$

which again we visualise to provide an idea of how the weights would vary with time (figure 9).

The log likelihood in this case will take into account all previous games but each term will be multiplied by some positive constant, since the ex-

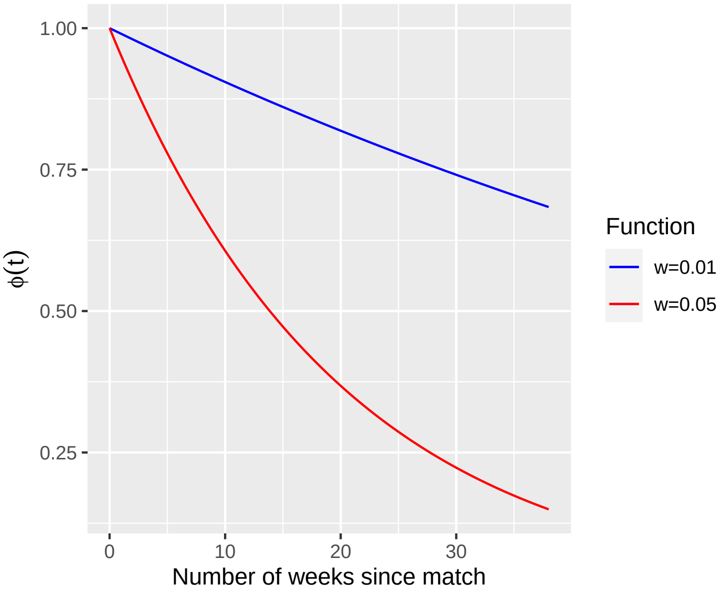


Figure 9: Exponential weighting functions

ponential is strictly positive. We know that concavity isn't affected by the multiplication of positive constants so the log likelihood will be strictly concave under the same assumptions derived in section 3.3. We don't know however if the function is upper compact and so we can't be sure that the MLE exists. Regardless we shall carry out the same procedure as in the dynamic model but with our exponential weights applied to the terms. Figure 10 shows how different values of w affect the model accuracy.

As before the red, green and blue points correspond to better, equal and worse accuracy, when compared with the dynamic model with no weighting. We know that $e^0 = 1$ so when $w = 0$ then all of the weights are equal to 1, which is equivalent to not weighting the terms. This is confirmed by the green point where $w = 0$. Also we have a fairly distinct optimal accuracy of 55.4% attained with values $w = 0.022, 0.024$. We break the tie by picking the value with the lower RPS, so $w = 0.022$ is our best choice.
Recall that our static model achieved this same accuracy of 55.4%. However the ultimate goal of building a model is to accurately predict new, unseen matches. This means that if we have to choose between the static model and the dynamic model with exponential weights, we should choose the model which we believe will generalise and perform well on new data. Therefore at the present we will consider the exponentially weighted model to be our best model, since the static model computes the MLEs once arbitrarily after matchweek 10 and uses these estimates for the rest of the season. Whereas the exponentially weighted model is able to appropriately weight the more recent matches and updates the teams' relative strengths as they evolve

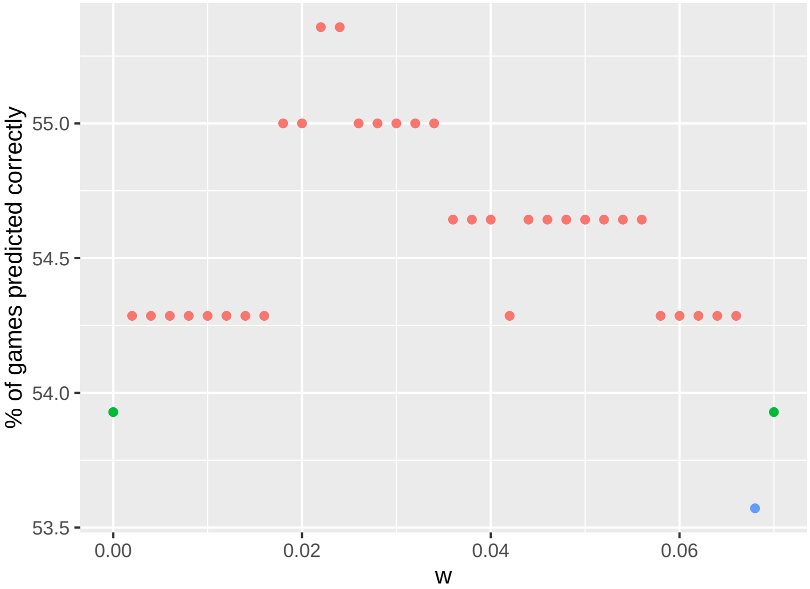


Figure 10: How w affects classification accuracy

through the season, which seems to more closely reflect the reality of a football season.

4 Bayesian Inference

4.1 Introduction

In the last section we looked at how we can infer the values of our model parameters in a frequentist framework, where the true parameter values, θ_0, are viewed as fixed but unknown quantities and the data as random realisations of the model with density $f(\cdot;\theta_0)$. However in the Bayesian framework we view the parameters as random objects and we will try to find the distribution of the parameters by conditioning on our set of data i.e. the parameters are random and the data is fixed, which in some sense flips the frequentist paradigm on its head. To see how we may infer the distributions of the parameters we look at a result known as Bayes' theorem, which we can derive using the definition of conditional probabilities. We have for events A and B

$$P(A|B)P(B) = P(A \cap B)$$

and similarly

$$P(B|A)P(A) = P(B \cap A).$$

Trivially $P(A \cap B) = P(B \cap A)$ so we can equate the terms on the left hand side and rearrange to obtain Bayes' theorem,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

Note that Bayes' theorem also holds for continuous random variables, which can be shown with an analogous proof using the definition of conditional densities. This is useful since our parameters are continuous and so we can use Bayes' theorem to find the posterior distribution of our parameters. Using θ to denote our parameters and $x = (x_1, ..., x_n)$ our data, then we have

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}.$$

The term on the left, $P(\theta|x)$, we refer to as the posterior distribution and represents the distribution of the parameters conditional on the observed data, and is what we want to find. Hence to figure out if we can calculate the posterior distribution let's take a look at the terms on the right hand side. The first term $P(x|\theta)$ is the probability of the data given θ which we recognise from section 3 to be nothing but the likelihood function. The next term $P(\theta)$ looks similar to the posterior distribution but it isn't conditional on the data, so this term denotes the distribution of the parameters before we have seen the data. This is referred to as the prior distribution which we specify to encode our prior beliefs about the parameters. Note that if we don't really have any prior beliefs about the parameters we can set uninformative priors on them. Hence we are always able to specify a prior distribution and we know how to compute the likelihood function, so no problems so far. The final term which we need to compute $P(x)$ is the normalising constant. We know that a valid probability distribution must integrate to 1 and the posterior distribution is no exception to this rule. To ensure the right hand side integrates to 1 we know that the normalising constant $P(x)$ must be equal to the integral of the numerator with respect to θ, that is,

$$P(x) = \int P(x|\theta)P(\theta)d\theta.$$

Computing this integral could be problematic since in our case θ is a vector of 21 parameters and so computing the high-dimensional integral isn't feasible. The integral will likely have no closed form solution so we could instead perform Monte Carlo integration and we will investigate how this method scales with dimension. To carry out Monte Carlo integration we can sample points uniformly from a region which we know encloses the hypersurface given by the integrand, and we can approximate the integral by the proportion of these points which fall under the hypersurface multiplied by the area of the region we sampled the points in. As an example, let us try to estimate the area of a circle with radius 1. We can enclose this circle with a square with sides of length 2 (figure 11).

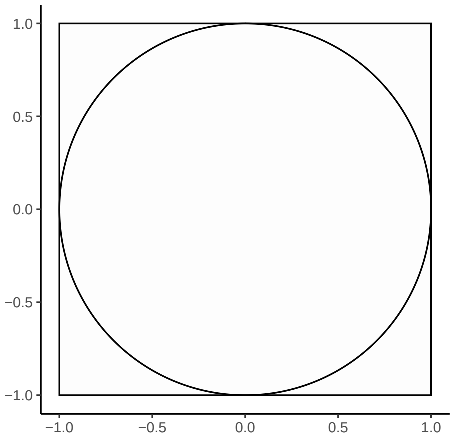


Figure 11: Circle of radius 1 enclosed by a 2x2 square

We then sample n points uniformly from within the 2x2 square, where we sample a point by generating $x, y \sim U[-1, 1]$ and then the coordinates of the point are given by (x, y). Doing this $n = 500$ times obtains something that looks similar to figure 12.

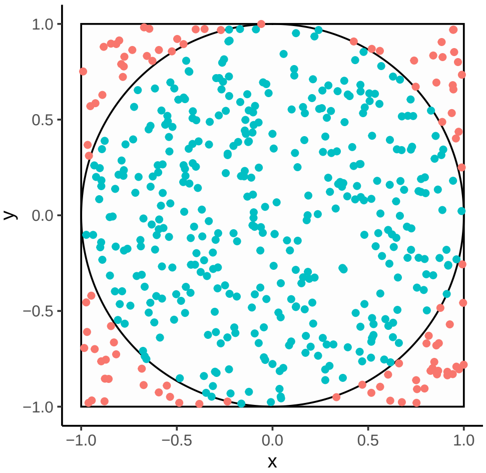


Figure 12: 500 points sampled uniformly at random within the square

The red points fall outside the circle and the blue points fall inside. Since these points are equally likely to fall anywhere within the square we therefore expect the proportion of points that fall within the circle to be similar to the ratio of the area of the circle to the area of the square. So we expect that $\frac{\text{area of circle}}{\text{area of square}} \approx Z$, where Z is the proportion of points that are inside the circle. Of course we know the area of the square is 4 allowing us to approximate the area of the circle.

The procedure works fine since the number of dimensions is low. However now let us consider the same problem but in 21 dimensions. The volume of a 21-dimensional hypersphere, s, of radius 1 is

$$\mathrm{vol}(s) = \frac{\pi^{\frac{21}{2}}}{\Gamma(\frac{21}{2}+1)}$$

where $\Gamma(\cdot)$ is the gamma function and we can similarly enclose s in a 21-dimensional hypercube with sides of length 2. The volume of this hypercube is 2^{21} and calculating the ratio of the volumes of the hypersphere and the hypercube gives

$$\frac{\mathrm{vol}(s)}{2^{21}} = \frac{\pi^{\frac{21}{2}}}{\Gamma(\frac{21}{2}+1)2^{21}} \approx 7 \times 10^{-9}.$$

This means that the probability of one of our random points landing in the hypersphere is also around 7×10^{-9}. Trying to find a Monte Carlo estimate here the same as before would prove unsuccessful since the event we are interested in happening, happens so rarely. As a result the relative variance of the Monte Carlo estimate in this scenario scales exponentially with the dimension and so the number of samples we would require to get a good estimate is likely to be unfeasible.

Unfortunately, this phenomenon is not unique to spheres and cubes. Our posterior distribution $P(\theta|x) \propto P(x|\theta)P(\theta)$, given that it is 21-dimensional, will also only take up a very small proportion of the entire parameter space and will suffer from the same problem, referred to as the curse of dimensionality. Rather than trying to find the normalising constant, instead we will try to obtain samples from the posterior distribution. There exist sampling methods such as rejection sampling which use similar ideas to what we saw in the circle example. However these methods also tend not to scale well with dimension so we must think of a method which allows us to focus more of our attention on the region of interest, allowing us to reduce the effect of the curse of dimensionality.

4.2 Markov chains

We have suggested that we need a method of sampling from our posterior distribution which allows us to focus more on the relatively small region of interest i.e. the region of space where the mode of the posterior distribution is contained is minuscule compared to the entire parameter space. To do this we will use Markov chains whose state space is $\mathbb{R}^d$.

Definition 1. *A sequence* $\left(X^{(t)}\right)_{t\geq 0}$ *is a Markov chain if for* $X^{(t+1)} \in \mathbb{R}^d$ *and* $A \subset \mathbb{R}^d$ *we have*

$$\begin{aligned}\mathbb{P}\left(X^{(t+1)} \in A | X^{(0)} = x_0, ..., X^{(t)} = x_t\right) &= \mathbb{P}\left(X^{(t+1)} \in A | X^{(t)} = x_t\right) \\ &= K(x_t, A)\end{aligned}$$

where K is the transition kernel of the Markov chain, and for each x, $K(x, \cdot)$ *is a probability distribution.*

The important property of a Markov chain is that the next value in the sequence depends only on the last state. For example, imagine a random walk on the $\mathbb{R}$ such that starting from some $X^{(0)} \in \mathbb{R}$, we have for all $t \geq 0$

$$X^{(t+1)} \sim \mathcal{N}\left(X^{(t)}, 1\right).$$

We can see that $X^{(t+1)}$ only depends on $X^{(t)}$ for all t and hence this random walk would create a Markov chain. To obtain samples from our posterior then, we could consider performing a random walk on the parameter space. If we use a random walk similar to the above example then each successive value in the chain will be close to the previous value which should allow us to focus more on our region of interest as we wanted. However, we can't just perform a random walk and expect that the resulting values in the chain are distributed according to our posterior distribution. To check whether this is the case we turn to the invariant distribution of the Markov chain.

Definition 2. *Let K be a transition kernel for some Markov chain. A probability distribution* π *is an invariant distribution for K if for all* $A \subset \mathbb{R}^d$

$$\int_{\mathbb{R}^d} \pi(x) K(x, A) dx = \int_A \pi(x) dx.$$

This tells us that if π is an invariant distribution for K and $X^{(t)} \sim \pi$ for some t, then all subsequent values $X^{(t+1)}, X^{(t+2)}, ...$ are distributed according to π as well. So we can implement a random walk as before, but we must ensure that our posterior distribution is the invariant distribution for the transition kernel of the Markov chain. If this is the case then once the Markov chain has converged to its invariant distribution, all subsequent values in the chain will be distributed according to the posterior. Therefore after convergence we can generate another n values in the chain, giving us n (not necessarily independent) samples from our posterior distribution. Note that it isn't guaranteed that such a Markov chain will converge to its invariant distribution but here we don't worry about conditions that ensure such convergence. One last definition:

Definition 3. *K is reversible with respect to* π *if for all* $A, B \subset \mathbb{R}^d$

$$\int_A \pi(x) K(x, B) dx = \int_B \pi(x) K(x, A) dx.$$

When showing that π is invariant for some transition kernel K, it may be easier to instead show that K is reversible with respect to π. We may do this since it is known that if K is reversible with respect to π, then π is invariant for K.

4.3 Metropolis-within-Gibbs algorithm

Here we will show the Metropolis-within-Gibbs algorithm for two dimensions even though for our purposes we will be interested in sampling from a distribution with 21 dimensions. This is just to keep things simple and rest assured all the same ideas will hold in higher dimensions.

Let π be our target distribution on $\mathbb{R}^2$, and q_1 and q_2 be two transition kernels on $\mathbb{R}$.

Algorithm 1 Metropolis-within-Gibbs

```
Initialise from some $X^{(0)} \in \mathbb{R}^2$, then iterate for t $\geq 1$
    1. Draw $X_1 \sim q_1(X_1^{(t-1)}, \cdot)$
    2. Compute
           $\alpha_1(X_1|X_1^{(t-1)}) = \min\left\{1, \frac{\pi(X_1, X_2^{(t-1)}) q_1(X_1, X_1^{(t-1)})}{\pi(X_1^{(t-1)}, X_2^{(t-1)}) q_1(X_1^{(t-1)}, X_1)}\right\}$
    3. With probability $\alpha_1(X_1|X_1^{(t-1)})$ accept the proposed value and set
       $X_1^{(t)} = X_1$, otherwise set $X_1^{(t)} = X_1^{(t-1)}$.
    4. Draw $X_2 \sim q_2(X_2^{(t-1)}, \cdot)$
    5. Compute
           $\alpha_2(X_2|X_2^{(t-1)}) = \min\left\{1, \frac{\pi(X_1^{(t)}, X_2) q_2(X_2, X_2^{(t-1)})}{\pi(X_1^{(t)}, X_2^{(t-1)}) q_2(X_2^{(t-1)}, X_2)}\right\}$
    6. With probability $\alpha_2(X_2|X_2^{(t-1)})$ set $X_2^{(t)} = X_2$, otherwise set $X_2^{(t)} =$
       $X_2^{(t-1)}$.
```

Proposition 1. *The transition kernel for the Metropolis-within-Gibbs algorithm has π as an invariant distribution.*

Proof. We will first show that, for $x, x' \in \mathbb{R}^2$ the transition kernel for the algorithm is of the form

$$K(x, dx') = K_1(x_1, dx'_1) K_2(x_2, dx'_2)$$

where K_1 is a transition kernel on $\mathbb{R}$ that has $\pi(\cdot|x_2)$ as invariant distribution and K_2 is a transition kernel on $\mathbb{R}$ that has $\pi(\cdot|x_1)$ as invariant distribution. The fact that the transition kernel for the algorithm can be split into two transition kernels is straightforward since the algorithm updates each component separately. To find the density of K_1 we notice that the event of the next state being in some set $A \subset \mathbb{R}$ given that the current state is x_1 can either happen if the proposed value is in A and is accepted, or if the current state x_1 is already in A and the proposed value is rejected. We can write

$$K_1(x_1, A) = \int_A q_1(x_1, x_1')\alpha_1(x_1'|x_1)dx_1' + \mathbb{1}\{x_1 \in A\}[1 - a(x_1)]$$

where $\mathbb{1}\{x_1 \in A\} = 1$ if $x_1 \in A$ and 0 otherwise, and
$a(x_1) = \int_{\mathbb{R}} q_1(x_1, x_1')\alpha_1(x_1'|x_1)dx_1'$. We know that if a transition kernel is reversible with respect to π, then π is an invariant distribution for that transition kernel. Hence to show that $\pi(\cdot|x_2)$ is an invariant distribution for K_1 it suffices to show that K_1 is reversible with respect to $\pi(\cdot|x_2)$, that is, we want to show that

$$\int_A \pi(x_1|x_2)K_1(x_1, B)dx_1 = \int_B \pi(x_1|x_2)K_1(x_1, A)dx_1.$$

Plugging K_1 into the left hand side gives

$$\begin{aligned}
&\int_A \pi(x_1|x_2)\left(\int_B q_1(x_1, x_1')\alpha_1(x_1'|x_1)dx_1' + \mathbb{1}\{x_1 \in B\}[1 - a(x_1)]\right)dx_1 \\
&= \int_A \pi(x_1|x_2)\int_B q_1(x_1, x_1')\alpha_1(x_1'|x_1)dx_1'dx_1 + \int_A \pi(x_1|x_2)\mathbb{1}\{x_1 \in B\}[1 - a(x_1)]dx_1 \\
&= \int_A\int_B \pi(x_1|x_2)q_1(x_1, x_1')\alpha_1(x_1'|x_1)dx_1'dx_1 + \int_{A\cap B} \pi(x_1|x_2)[1 - a(x_1)]dx_1.
\end{aligned}$$

The same reasoning yields for the right hand side

$$\int_B\int_A \pi(x_1|x_2)q_1(x_1, x_1')\alpha_1(x_1'|x_1)dx_1'dx_1 + \int_{B\cap A} \pi(x_1|x_2)[1 - a(x_1)]dx_1.$$

The second terms of both sides are equal since $A \cap B = B \cap A$ and for the first terms to be equal we need to be able to swap x_1 and x_1' in the integrand. More specifically, we require

$$\pi(x_1|x_2)q_1(x_1, x_1')\alpha_1(x_1'|x_1) = \pi(x_1'|x_2)q_1(x_1', x_1)\alpha_1(x_1|x_1').$$

Working from left hand side to right, we have

$$\begin{aligned}
&\pi(x_1|x_2)q_1(x_1,x_1')\alpha_1(x_1'|x_1)\\
&=\pi(x_1|x_2)q_1(x_1,x_1')\min\left\{1,\frac{\pi(x_1',x_2)q_1(x_1',x_1)}{\pi(x_1,x_2)q_1(x_1,x_1')}\right\}\\
&=\pi(x_1|x_2)q_1(x_1,x_1')\min\left\{1,\frac{\pi(x_1'|x_2)\pi(x_2)q_1(x_1',x_1)}{\pi(x_1|x_2)\pi(x_2)q_1(x_1,x_1')}\right\}\\
&=\min\left\{\pi(x_1|x_2)q_1(x_1,x_1'),\pi(x_1'|x_2)q_1(x_1',x_1)\right\}\\
&=\pi(x_1'|x_2)q_1(x_1',x_1)\min\left\{\frac{\pi(x_1|x_2)q_1(x_1,x_1')}{\pi(x_1'|x_2)q_1(x_1',x_1)},1\right\}\\
&=\pi(x_1'|x_2)q_1(x_1',x_1)\min\left\{\frac{\pi(x_1,x_2)\pi(x_2)q_1(x_1,x_1')}{\pi(x_1',x_2)\pi(x_2)q_1(x_1',x_1)},1\right\}\\
&=\pi(x_1'|x_2)q_1(x_1',x_1)\min\left\{1,\frac{\pi(x_1,x_2)q_1(x_1,x_1')}{\pi(x_1',x_2)q_1(x_1',x_1)}\right\}\\
&=\pi(x_1'|x_2)q_1(x_1',x_1)\alpha_1(x_1|x_1').
\end{aligned}$$

Hence $\pi(\cdot|x_2)$ is an invariant distribution for K_1 and an identical argument shows that $\pi(\cdot|x_1)$ is an invariant distribution for K_2. It remains to show that π is an invariant distribution for the transition kernel K, in particular, we must show that $\int_{\mathbb{R}^2}\pi(x)K(x,dx')dx=\pi(x')dx'$ for all $x'\in\mathbb{R}^2$. Let $x'\in\mathbb{R}^2$, we have

$$\begin{aligned}
\int_{\mathbb{R}^2}\pi(x)K(x,dx')dx&=\int_{\mathbb{R}^2}\pi(x_1,x_2)K_1(x_1,dx_1')K_2(x_2,dx_2')dx_1dx_2\\
&=\int_{\mathbb{R}^2}\pi(x_1|x_2)\pi(x_2)K_1(x_1,dx_1')K_2(x_2,dx_2')dx_1dx_2\\
&=\int_{\mathbb{R}}\left(\int_{\mathbb{R}}\pi(x_1|x_2)K_1(x_1,dx_1')dx_1\right)\pi(x_2)K_2(x_2,dx_2')dx_2\\
&=\int_{\mathbb{R}}\pi(x_1'|x_2)\pi(x_2)K_2(x_2,dx_2')dx_1'dx_2\\
&=\int_{\mathbb{R}}\pi(x_1',x_2)K_2(x_2,dx_2')dx_1'dx_2\\
&=\int_{\mathbb{R}}\pi(x_2|x_1')\pi(x_1')K_2(x_2,dx_2')dx_1'dx_2\\
&=\pi(x_1')dx_1'\int_{\mathbb{R}}\pi(x_2|x_1')K_2(x_2,dx_2')dx_2\\
&=\pi(x_1')dx_1'\pi(x_2'|x_1')dx_2'=\pi(x_1',x_2')dx_1'dx_2'=\pi(x')dx',
\end{aligned}$$

as required. □

Therefore from this algorithm we will obtain a Markov chain $(X_1^{(t)}, X_2^{(t)})_{t\geq 0}$ which has π as its invariant distribution. This means that once our Markov chain has converged to its invariant distribution, then we can treat all subsequent values as samples from our target distribution π, from which we can calculate any quantities of interest such as the expectation of π. Of course since states of a Markov chain depend on their previous state our samples will be correlated, which will increase the variance of our estimate of the expectation compared to i.i.d samples. However we will also have a reduction in the variance of our estimates due to the fact that we exploit Markov chains to focus on the region of interest allowing us to obtain many more samples, which will generally outweigh the negatives associated with the correlation. As well as scaling relatively well with dimension, this algorithm has another very important property. Imagine that we only know the target distribution π up to a constant of proportionality i.e. $\pi = c\gamma$ where c is some constant and γ is the unnormalised distribution. Then the acceptance probabilities from our algorithm are

$$\begin{aligned}\alpha_1(X_1|X_1^{(t-1)}) &= \min\left\{1, \frac{\pi(X_1, X_2^{(t-1)})q_1(X_1, X_1^{(t-1)})}{\pi(X_1^{(t-1)}, X_2^{(t-1)})q_1(X_1^{(t-1)}, X_1)}\right\}\\ &= \min\left\{1, \frac{c\gamma(X_1, X_2^{(t-1)})q_1(X_1, X_1^{(t-1)})}{c\gamma(X_1^{(t-1)}, X_2^{(t-1)})q_1(X_1^{(t-1)}, X_1)}\right\}\\ &= \min\left\{1, \frac{\gamma(X_1, X_2^{(t-1)})q_1(X_1, X_1^{(t-1)})}{\gamma(X_1^{(t-1)}, X_2^{(t-1)})q_1(X_1^{(t-1)}, X_1)}\right\}\end{aligned}$$

which doesn't depend on the normalising constant c. This is useful for us since we are trying to obtain samples from our posterior distribution which takes the form

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

where we don't know the normalising constant $P(x)$, but as we have just seen we don't need to and we can implement algorithm 1 anyway.

One last note on our algorithm is how we choose our proposal distributions. A popular choice is to let the proposal distributions be some symmetric distribution, such as the normal distribution. For example, let the proposal $q_1(X_1, X_1^{(t-1)})$ be given by a normal distribution with mean $X_1^{(t-1)}$ and standard deviation σ, where σ will determine roughly how close the proposed value is to the previous value. The normal distribution is symmetric since

$$\begin{aligned}q_1(X_1, X_1^{(t-1)}) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}\left(X_1 - X_1^{(t-1)}\right)^2} &= \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}\left(X_1^{(t-1)} - X_1\right)^2}\\ &= q_1(X_1^{(t-1)}, X_1).\end{aligned}$$

This is helpful since it simplifies the acceptance probability, which becomes

$$\begin{aligned}\alpha_1(X_1|X_1^{(t-1)}) &= \min\left\{1, \frac{\pi(X_1, X_2^{(t-1)})q_1(X_1, X_1^{(t-1)})}{\pi(X_1^{(t-1)}, X_2^{(t-1)})q_1(X_1^{(t-1)}, X_1)}\right\}\\ &= \min\left\{1, \frac{\pi(X_1, X_2^{(t-1)})}{\pi(X_1^{(t-1)}, X_2^{(t-1)})}\right\}.\end{aligned}$$

Hence setting all our proposals to be normal, we make the algorithm slightly easier to implement as well as making it more interpretable. We can now see that the algorithm is essentially a random walk on each of the parameters, where the proposed value is always accepted if it is more probable than the previous state, since in that case $\pi(X_1, X_2^{(t-1)})/\pi(X_1^{(t-1)}, X_2^{(t-1)}) \geq 1$ giving an acceptance probability of 1. Additionally, if the proposed value is less probable than the previous state then it is accepted with probability $\pi(X_1, X_2^{(t-1)})/\pi(X_1^{(t-1)}, X_2^{(t-1)})$.

4.4 Diagnostics

4.4.1 Plots

Now we have an effective way to sample from our posterior distribution, but before we implement the Metropolis-within-Gibbs algorithm we must understand how to choose the standard deviations of our proposal distributions and set prior distributions on our parameters. Remember that we have 21 parameters in our model, the first 20 of which are the log strength terms, λ_i, and we reparameterise the last parameter such that $\phi = \log(\nu)$ since $\nu \geq 0$. We don't have much prior knowledge about the relative strengths of the teams or the prevalence of draws so we just set all of our priors to be independent $\mathcal{N}(0, 1)$ distributions. Naturally, all the log strength terms are on a similar scale and so it makes sense to use the same standard deviation for their proposal distributions. This makes the implementation a bit easier since now we only need to specify 2 different standard deviations, one for the log strength parameters, call it σ_1, and one for the parameter ϕ, call it σ_2. We will use all of the data from the 16/17 Premier League season to construct our likelihood function (in practice we work in the log domain as before) and now we can implement the algorithm to sample from our posterior distribution. We show trace plots for the parameters λ_7 and ϕ for different values of the standard deviations.

In figures 13 and 14, we can see that there are some fairly long horizontal parts of the plots. This tells us that many successive values in the Markov chain are the same implying that all of the corresponding proposed values were rejected, likely due to the standard deviations being too large. The result of this is a high autocorrelation since many of the values will be identical. In fact the autocorrelations for these chains are 0.96 and 0.97 respectively.

In figures 15 and 16 we have the opposite problem of the standard deviations being too low. This means that every proposed value will be very

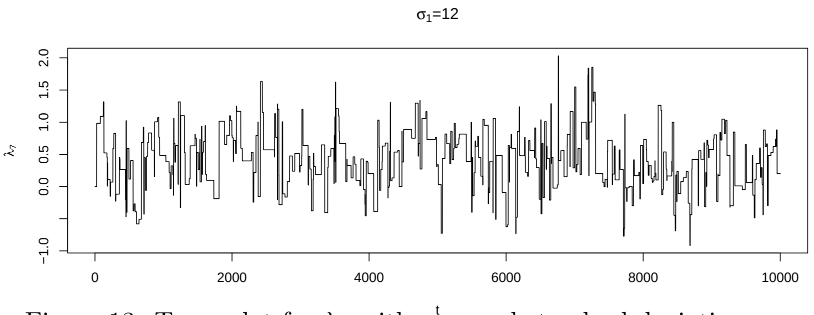


Figure 13: Trace plot for λ_7 with proposal standard deviation $\sigma_1 = 12$

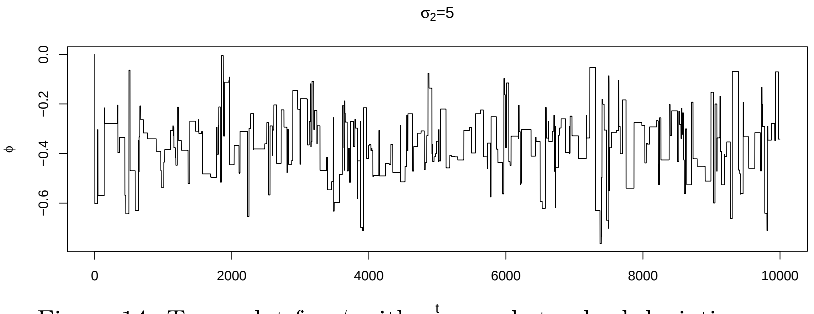


Figure 14: Trace plot for ϕ with proposal standard deviation $\sigma_2 = 5$

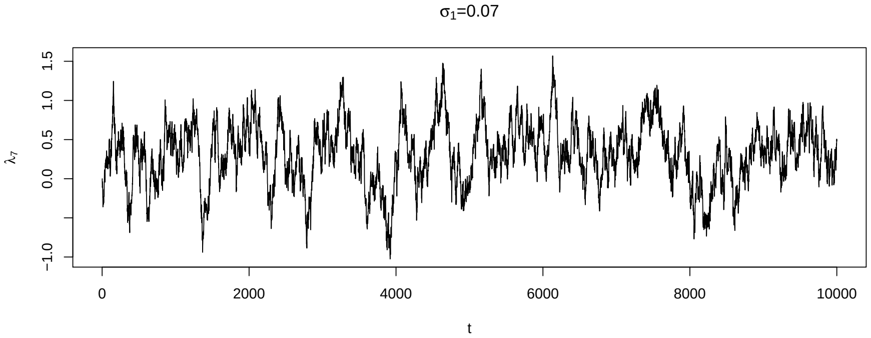


Figure 15: Trace plot for λ_7 with proposal standard deviation $\sigma_1 = 0.07$

close to the previous value in the chain, and again this will cause successive values to be highly correlated. The autocorrelation associated with these trace plots is 0.99 in both cases.

We have seen examples of when the standard deviations are too high or too low, so now let us see examples of what we generally want our trace plots to look like.

In figures 17 and 18 we can see that the chain is exploring the parameter space much more efficiently, and we are rewarded with lower autocorrelations

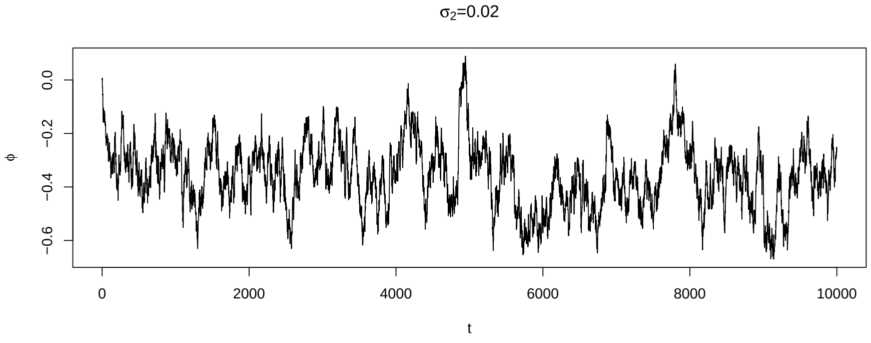


Figure 16: Trace plot for ϕ with proposal standard deviation $\sigma_2 = 0.02$

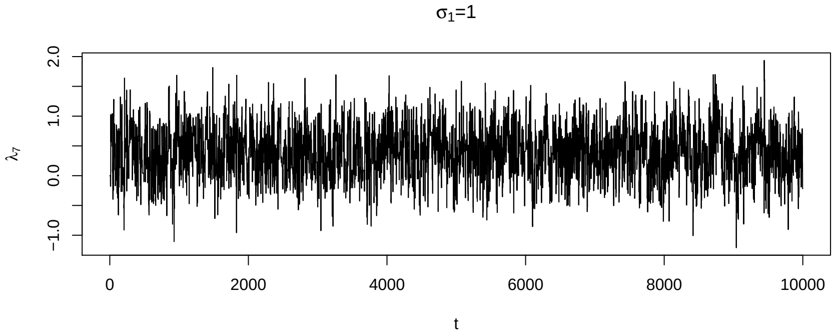


Figure 17: Trace plot for λ_7 with proposal standard deviation $\sigma_1 = 1$

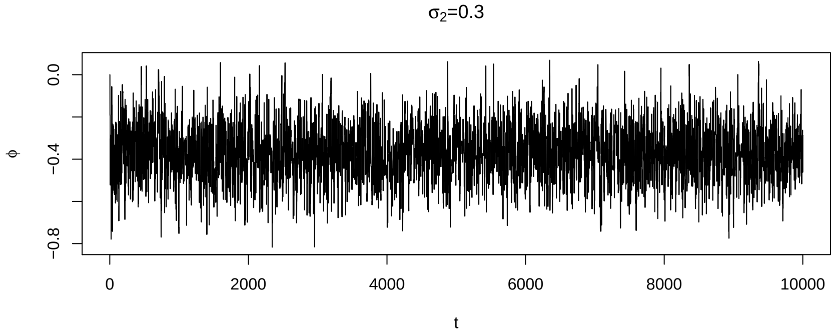


Figure 18: Trace plot for ϕ with proposal standard deviation $\sigma_2 = 0.3$

of 0.69 and 0.66 respectively.

Another important consideration is whether or not the chain has actually converged to its invariant distribution. Recall that algorithm 1 works because it produces a Markov chain which has our target distribution as invariant, however we can't be sure that the chain will converge to this distribution in a reasonable amount of time. However looking at the last set of plots it looks as though both chains have converged straight away, since they seem to consistently move around the same set of values. To make it clear what we mean by this we show an example of a chain which doesn't converge as quickly (figure 19).

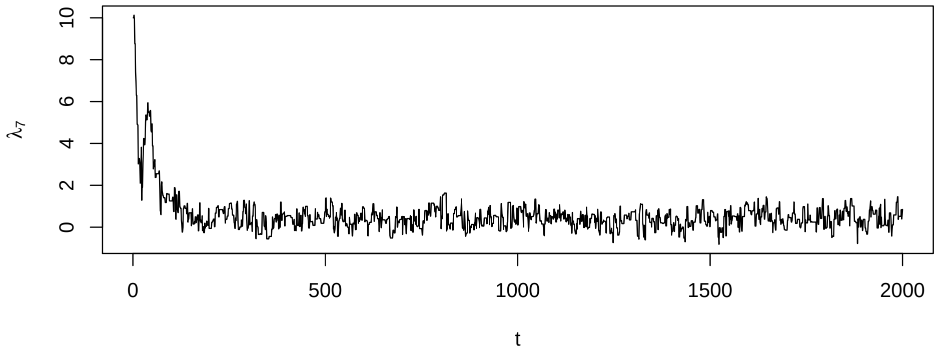


Figure 19: Slow convergence of the Markov chain

We can see that in roughly the first 250 iterations there is no real pattern to the chain. After around $t = 250$ we then start to see a consistent pattern, which tells us that the chain has likely converged. To confirm this mathematically we can perform a Kolmogorov-Smirnov test, which is a non-parametric test of whether two samples are from the same distribution. If this chain has indeed converged to its invariant distribution, then the first 1000 values and the second 1000 values should appear as if they are from the same distribution. Performing a K-S test comparing the first and second 1000 values yields a p-value of approximately 4×10^{-8}. This means that there is substantial evidence to reject the null hypothesis that the two samples are from the same distribution, agreeing with what we see in figure 19. We deal with this by simply removing the part of the chain before convergence has taken place. Discarding the first 300 values gives us the reduced chain in figure 20.

This chain looks as though it has converged. Again we can confirm this by performing a K-S test comparing the first 850 and second 850 values of the reduced chain. Such a test yields a p-value of 0.30, hence we don't have much reason to reject that the two samples are from different distributions and we conclude that the reduced chain has converged.

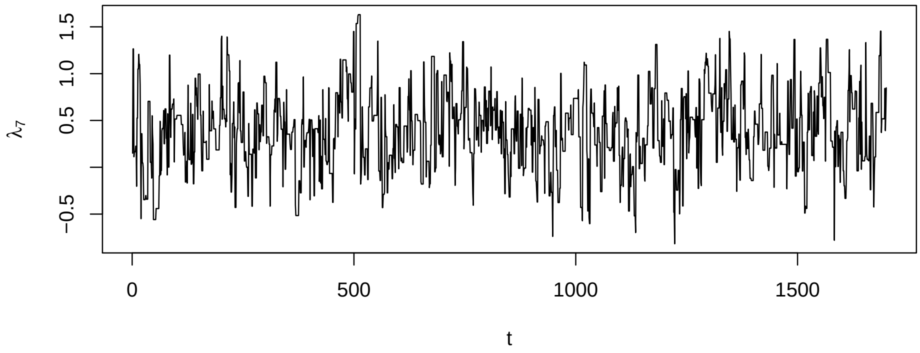


Figure 20: Markov chain after convergence

4.4.2 Acceptance rates

We have seen that we need to appropriately choose the standard deviations of the proposal distributions to get a trace plot which looks as though the chain is efficiently searching the parameter space. For a more objective measure of how to tune the standard deviations we turn to the acceptance rates i.e. the proportion of proposed values which are accepted. We saw that a standard deviation which is too low always proposes values close to the previous value and so we end up accepting most of the proposed values. Also a standard deviation which is too high proposes values which are too far from the previous value and more likely to be rejected, so we get a low acceptance rate. Hence we don't want an acceptance rate that is close to 1 or close to 0, so the question is, what is the optimal acceptance rate? Roberts et al. (1997) show that the chain searching the parameter space can be thought of as a diffusion process, and the speed of diffusion should be maximised. In the case of a standard normal target distribution and a normal proposal distribution the speed of diffusion is given by

$$h(\phi) = 2\phi^2\Phi\left(\frac{-\phi}{2}\right)$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal and $\phi = \sigma\sqrt{d}$ with σ the standard deviation of the proposal and d the number of dimensions of the proposal. Note that although our model is in 21 dimensions, the Metropolis-within-Gibbs algorithm updates the parameters one at a time so we are interested in the case $d = 1$. We show the function $h(\phi)$ in figure 21.

We can see that the speed of diffusion is maximised when $\phi = 2.38$, which corresponds to a proposal standard deviation of 2.38 also. Note however that the optimal standard deviation of the proposal distribution will vary depending on the scale of the target distribution. Therefore we want to

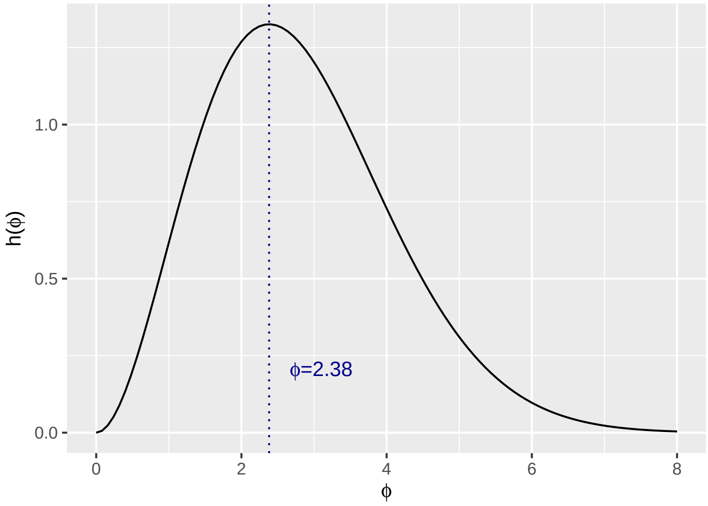


Figure 21: Speed of diffusion

find the optimal average acceptance rate which we expect will be relatively similar for different target distributions. We want to find the expectation of the acceptance probability, $\alpha(y|x) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}$, when our normal proposal distribution $q(x, y)$ has optimal standard deviation $\sigma = 2.38$ and π is our standard normal target distribution. We have

$$\begin{aligned}\mathbb{E}[\alpha(y|x)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \alpha(y|x)\pi(x)q(x, y)\,dx\,dy\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}\pi(x)q(x, y)\,dx\,dy\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \min\{\pi(x), \pi(y)\}q(x, y)\,dx\,dy\end{aligned}$$

We can approximate this integral by Monte Carlo integration since we are in a small number of dimensions. Note that the value of this integral is just the volume under the surface given by $f(x, y) = \min\{\pi(x), \pi(y)\}q(x, y)$, shown in figure 22.

Recall that to approximate the volume under the surface we can sample uniformly from a region which we know encloses the surface. Although $x, y \in (-\infty, \infty)$ to make the problem easier we will only find the volume under the surface for $x, y \in (-10, 10)$. We can do this since the largest point of the surface outside this region is approximately 6×10^{-39} and so the vast majority of the volume will be contained within this region. To find the height of the box in which we will enclose the surface, we need to find the maximum and minimum values of the function $f(x, y)$. The function is effectively a product of probability densities so we know the surface will

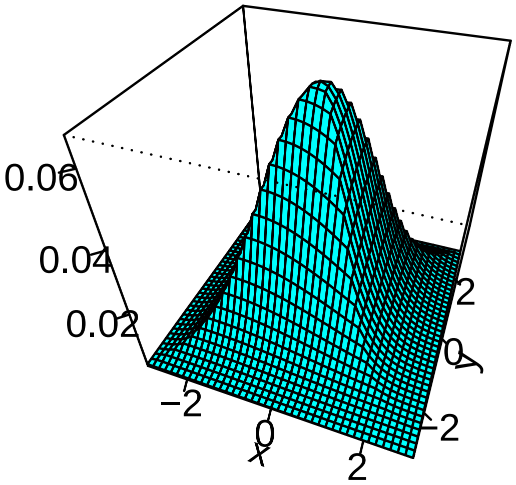


Figure 22: Plot of $f(x,y) = \min\{\pi(x), \pi(y)\}q(x,y)$

not take values less than 0. Additionally it can be seen in the above plot that the maximum value is somewhere around 0.06. To find the exact value notice that to maximise $\min\{\pi(x), \pi(y)\}$ we set $x = y = 0$ since a standard normal has highest density at its mean value 0. Also $q(x,y)$ will attain its maximum value at $y = x$ again since it is normal and has highest density at its mean, x. It is clear then that the function $f(x,y)$ will be maximised at $x = y = 0$, at a value of $f(0,0) = \min\{\pi(0), \pi(0)\}q(0,0) = 0.0668$. Now we know we can contain the surface (or the portion of the surface under which most of the volume lies) within a cuboid, C, the region defined by

$$C = \{(x,y,z) : -10 \leq x \leq 10, -10 \leq y \leq 10, 0 \leq z \leq 0.067\}.$$

All that remains is to sample uniformly from this cuboid many times and find the proportion of these points which fall underneath the surface. Doing this 10^7 times we find that a proportion of roughly 0.0166 points fall under the surface. Since the expected number of points that fall under the surface is equal to the ratio of the volume under the surface and the volume of C, then we can estimate the volume under the surface by multiplying 0.0166 by the volume of C. Hence our estimate of the volume under the surface is $0.0166 \times 20^2 \times 0.067 \approx 0.44$.

We can also find the optimal acceptance rate by calculating the expected value of the acceptance probability analytically. Note that the Metropolis-within-Gibbs algorithm with a one-dimensional target distribution is known as the Metropolis-Hastings algorithm and consider the following result.

Proposition 2. *Suppose we are implementing the Metropolis-Hastings algorithm with a standard normal target distribution and a normal proposal distribution with standard deviation σ. Then with $g(\cdot)$ the target density, X distributed according to the target distribution and Y distributed according to the proposal distribution, the acceptance probability of the algorithm has expected value*

$$\mathbb{E}\left[\min\left\{1, \frac{g(Y)}{g(X)}\right\}\right] = \frac{2}{\pi}\arctan\left(\frac{2}{\sigma}\right).$$

Proof. Suppose X is distributed according to the standard normal and Y is conditionally normal with mean X and standard deviation σ. Denoting the proposal distribution $q(\cdot,\cdot)$ and the target distribution $g(\cdot)$, we have

$$\begin{aligned}\mathbb{E}\left[\min\left\{1, \frac{g(Y)}{g(X)}\right\}\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \min\left\{1, \frac{g(y)}{g(x)}\right\} g(x)q(x,y)\,dx\,dy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \min\left\{g(x), g(y)\right\} q(x,y)\,dx\,dy.\end{aligned}$$

We know that $g(-x) = g(x)$ and $q(-x,-y) = q(x,y)$ and therefore our integrand satisfies $\min\{g(-x), g(-y)\}\, q(-x,-y) = \min\{g(x), g(y)\}\, q(x,y)$ i.e. the integrand is symmetric about the origin. Let us exploit this fact by partitioning the domain into the four sub-domains which are created by the lines $y = x$ and $y = -x$. Due to the symmetry, an equivalent way of evaluating the integral is to integrate over one of these sub-domains and multiply by 4, we have

$$\begin{aligned}I &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \min\left\{g(x), g(y)\right\} q(x,y)\,dx\,dy \\ &= 4\int_{0}^{\infty}\int_{-x}^{x} \min\left\{g(x), g(y)\right\} q(x,y)\,dy\,dx.\end{aligned}$$

Now notice that we are integrating y between x and -x, so in this region y satisfies $-x \leq y \leq x \iff y \leq |x|$. We know that a standard normal density gets smaller at more extreme values and that in this region x is always a more extreme value than y, since $y \leq |x|$. It follows that in this region it is always the case that $g(x) \leq g(y)$ and so

$$\begin{aligned}I &= 4\int_{0}^{\infty}\int_{-x}^{x} \min\left\{g(x), g(y)\right\} q(x,y)\,dy\,dx \\ &= 4\int_{0}^{\infty}\int_{-x}^{x} g(x)q(x,y)\,dy\,dx \\ &= 4\int_{0}^{\infty}\int_{-x}^{x} \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2}\,dy\,dx \\ &= \frac{2}{\pi\sigma}\int_{0}^{\infty}\int_{-x}^{x} e^{-\frac{1}{2}x^2}e^{-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2}\,dy\,dx.\end{aligned}$$

To evaluate this integral we consider a transformation to polar coordinates given by $T(x, y) = (r\cos t, r\sin t)$. Before we can substitute in the new coordinates we must find the determinant of the Jacobian of the transformation T,

$$\begin{vmatrix} \frac{\partial r\cos t}{\partial r} & \frac{\partial r\cos t}{\partial t} \\ \frac{\partial r\sin t}{\partial r} & \frac{\partial r\sin t}{\partial t} \end{vmatrix} = \begin{vmatrix} \cos t & -r\sin t \\ \sin t & r\cos t \end{vmatrix} = r\cos^2 t + r\sin^2 t = r.$$

Then transforming coordinates gives

$$\begin{aligned}
I &= \frac{2}{\pi\sigma}\int_0^\infty \int_{-x}^{x} e^{-\frac{1}{2}x^2} e^{-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2}\, dy\, dx \\
&= \frac{2}{\pi\sigma}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} \int_0^\infty e^{-\frac{1}{2}r^2\cos^2 t} e^{-\frac{1}{2\sigma^2}(r\sin t - r\cos t)^2} r\, dr\, dt \\
&= \frac{2}{\pi\sigma}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} \int_0^\infty e^{-\frac{1}{2}r^2\cos^2 t - \frac{r^2}{2\sigma^2}(\sin t - \cos t)^2} r\, dr\, dt \\
&= \frac{2}{\pi\sigma}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} \left[\frac{1}{-\cos^2 t - \frac{1}{\sigma^2}(\sin t - \cos t)^2} e^{-\frac{1}{2}r^2\cos^2 t - \frac{r^2}{2\sigma^2}(\sin t - \cos t)^2}\right]_0^\infty dt \\
&= \frac{2}{\pi\sigma}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} \left(\frac{1}{-\cos^2 t - \frac{1}{\sigma^2}(\sin t - \cos t)^2}\right)\left(e^{-\infty} - e^0\right) dt \\
&= \frac{2}{\pi\sigma}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} \left(\frac{1}{\cos^2 t + \frac{1}{\sigma^2}(\sin t - \cos t)^2}\right) dt \\
&= \frac{2}{\pi\sigma}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} \left(\frac{\sec^2 t}{1 + \left(\frac{\tan t - 1}{\sigma}\right)^2}\right) dt.
\end{aligned}$$

Now consider the substitution $z = \arctan\left(\frac{\tan t - 1}{\sigma}\right)$. We have $\tan z = \frac{\tan t - 1}{\sigma}$ and differentiating both sides with respect to t gives $\sec^2 z \frac{dz}{dt} = \frac{\sec^2 t}{\sigma} \iff \sec^2 t\, dt = \sigma\sec^2 z\, dz$. The new limits are $z = \arctan\left(\frac{\tan(\frac{\pi}{4}) - 1}{\sigma}\right) = 0$ and $z = \arctan\left(\frac{\tan(-\frac{\pi}{4}) - 1}{\sigma}\right) = \arctan\left(-\frac{2}{\sigma}\right)$, and we have

$$\begin{aligned}
I = \frac{2}{\pi\sigma}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} \left(\frac{\sec^2 t}{1 + \left(\frac{\tan t - 1}{\sigma}\right)^2}\right) dt &= \frac{2}{\pi\sigma}\int_{\arctan\left(-\frac{2}{\sigma}\right)}^{0} \frac{\sigma\sec^2 z\, dz}{1 + \tan^2 z} \\
&= \frac{2}{\pi}\int_{\arctan\left(-\frac{2}{\sigma}\right)}^{0} \frac{\sec^2 z}{1 + \tan^2 z}\, dz.
\end{aligned}$$

Using the trigonometric identity $\sec^2 z = 1 + \tan^2 z$, we have

$$\begin{aligned}
I = \frac{2}{\pi}\int_{\arctan\left(-\frac{2}{\sigma}\right)}^{0} 1\, dz = \frac{2}{\pi}\left[z\right]_{\arctan\left(-\frac{2}{\sigma}\right)}^{0} &= \frac{2}{\pi}\left(0 - \arctan\left(-\frac{2}{\sigma}\right)\right) \\
&= \frac{2}{\pi}\arctan\left(\frac{2}{\sigma}\right)
\end{aligned}$$

since $\arctan(-x) = -\arctan(x)$. □

Recalling that the optimal standard deviation was $\sigma = 2.38$ we can find the average acceptance rate for this value is

$$\frac{2}{\pi} \arctan\left(\frac{2}{2.38}\right) \approx 0.44.$$

Therefore the optimal acceptance rate for a normal proposal and a standard normal target distribution is around 0.44. Of course our target distributions won't be standard normals but this result can give us some rough guidance on what to aim for.

4.4.3 Autocorrelation and effective sample size

We have mentioned that chains with higher autocorrelation are worse and here we will show a result that demonstrates why this is the case. Suppose that $X^{(0)}, X^{(1)}, ...$ is a stationary Markov chain with initial distribution equal to our target distribution π i.e. $X^{(0)} \sim \pi$. Now suppose that we wish to find the expectation of our target distribution, $\mathbb{E}[X^{(0)}]$, which we estimate by

$$\hat{\mu}_n = \frac{1}{n} \sum_{t=0}^{n-1} X^{(t)}.$$

Then under certain conditions which we won't discuss here, the Markov chain central limit theorem (Jones, 2004) tells us that

$$\hat{\mu}_n \approx \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$$

where μ is the true expectation of $X^{(0)}$ and
$\sigma^2 = \text{var}[X^{(0)}] + 2\sum_{k=1}^{\infty} \text{cov}[X^{(0)}, X^{(k)}]$. Explicitly writing out the variance of $\hat{\mu}_n$, we have

$$\text{var}\left[\frac{1}{n} \sum_{t=0}^{n-1} X^{(t)}\right] = \frac{1}{n}\text{var}[X^{(0)}] + \frac{2}{n} \sum_{k=1}^{\infty} \text{cov}[X^{(0)}, X^{(k)}]. \tag{6}$$

Note that the term on the left hand side is the variance of our estimate of the mean using the values of the Markov chain. To understand what the first term on the right hand side is, imagine we have independent and identically distributed (i.i.d) samples $Y_1, ..., Y_n$ from some distribution. The Monte Carlo estimate of the expectation of this distribution would simply be the sample mean, $\frac{1}{n}\sum_{i=1}^{n} Y_i$, and the variance of this estimate would be

$$\text{var}\left[\frac{1}{n} \sum_{i=1}^{n} Y_i\right] = \frac{1}{n^2}\text{var}[Y_1 + ... + Y_n] = \frac{n}{n^2}\text{var}[Y_1] = \frac{1}{n}\text{var}[Y_1].$$

Now comparing this to equation (6) we can see that $\frac{1}{n}\text{var}[X^{(0)}]$ is nothing but the variance of a Monte Carlo estimate of the expectation of π using i.i.d samples. Hence equation (6) is useful since it tells us precisely the relationship between the variance of an estimate that uses correlated samples of a Markov chain and the variance of an estimate that uses i.i.d samples. In our case the sum of covariances in the last term will always be positive and so the estimate using the Markov chain will always have a higher variance than the i.i.d estimate. In particular, the higher the autocorrelation of a chain is the higher these covariance terms will be, resulting in a comparably worse estimate from the Markov chain.

To get a better idea of how much these covariance terms negatively impact our estimate of the expectation, we follow Geyer (n.d.) and approximate our chain $\left(X^{(t)}\right)_{t\geq 0}$ as a stationary AR(1) process. This is an autoregressive model which is defined by

$$X^{(t+1)} = \rho X^{(t)} + \varepsilon_t$$

where ε_t is white noise with zero mean and constant variance, and $|\rho| < 1$ is the correlation between successive values. This model has the following useful property

$$\begin{aligned}
\mathrm{cov}[X^{(0)}, X^{(k)}] &= \mathrm{cov}[X^{(0)}, \rho X^{(k-1)} + \varepsilon_{t-1}] \\
&= \rho \ \mathrm{cov}[X^{(0)}, X^{(k-1)}] + \mathrm{cov}[X^{(0)}, \varepsilon_{t-1}] \\
&= \rho \ \mathrm{cov}[X^{(0)}, X^{(k-1)}] \\
&= \rho^2 \ \mathrm{cov}[X^{(0)}, X^{(k-2)}] \\
&= \ldots = \rho^k \mathrm{cov}[X^{(0)}, X^{(0)}] = \rho^k \mathrm{var}[X^{(0)}].
\end{aligned}$$

Combining this with (6) we obtain

$$\begin{aligned}
\mathrm{var}\left[\frac{1}{n}\sum_{t=0}^{n-1} X^{(t)}\right] &= \frac{1}{n}\mathrm{var}[X^{(0)}] + \frac{2}{n}\sum_{k=1}^{\infty}\mathrm{cov}[X^{(0)}, X^{(k)}] \\
&= \frac{1}{n}\mathrm{var}[X^{(0)}] + \frac{2}{n}\sum_{k=1}^{\infty}\rho^k \mathrm{var}[X^{(0)}] \\
&= \frac{1}{n}\mathrm{var}[X^{(0)}] + \frac{2\mathrm{var}[X^{(0)}]}{n}\sum_{k=1}^{\infty}\rho^k \\
&= \frac{1}{n}\mathrm{var}[X^{(0)}] + \frac{2\mathrm{var}[X^{(0)}]}{n}\frac{\rho}{1-\rho} \\
&= \frac{\mathrm{var}[X^{(0)}]}{n}\left[1 + \frac{2\rho}{1-\rho}\right] \\
&= \frac{\mathrm{var}[X^{(0)}]}{n}\left[\frac{1+\rho}{1-\rho}\right].
\end{aligned}$$

We can estimate ρ with the autocorrelation at lag 1 of our chain, that is, the correlation of the original chain, $\left(X^{(t)}\right)_{t\geq 0}$, and this same chain delayed by 1 time step. Now we can calculate approximately how much worse our estimate of the expectation is compared to the i.i.d case. Following on from the last result we can define the effective sample size, n_{eff}, of our chain to be

$$n_{eff} \approx n\frac{1-\rho}{1+\rho}$$

where n is the number of samples in our chain. This can be useful in practice since it tells us roughly the number of i.i.d samples that our chain is equivalent to. As a concrete example, we use the trace plot given in figure 17 from page 58, which had an autocorrelation of 0.69. Then calculating the effective sample size gives

$$n_{eff} \approx 10000\left(\frac{1-0.69}{1+0.69}\right) \approx 1800$$

and so this chain is roughly equal to 1800 i.i.d samples.

4.5 Simulated data

As we did for the MLE, we shall test our Bayesian inference technique on some simulated data. In particular we will use the same simulated data from before so we can see how similar the two methods are in this case. Using our diagnostics from earlier we find that a standard deviation for the proposals of the log strengths given by $\sigma_1 = 1$ and a standard deviation for the proposal of ϕ given by $\sigma_2 = 0.25$ seem to work well. Now we can estimate the values of the parameters with the means of our samples and we also report standard errors and 95% credible intervals for these estimates (table 15).

Recall that our maximum likelihood estimates for the strengths satisfied the constraint $\sum_{i=1}^{20} \pi_i = 1$ and in general the posterior means will not satisfy this constraint. To enable us to compare our estimates given by the posterior means with our maximum likelihood estimates we normalise the posterior means of the strength parameters subject to this constraint. This allows us to compare the two methods of inference in table 16.

The mean squared error in the maximum likelihood estimates was 0.00098 whereas the mean squared error for the (normalised) estimates given by the posterior means is 0.00056. Hence the Bayesian approach performs slightly better here in terms of mean squared error but the difference is small.

Parameter	Posterior mean	Standard error	95% Credible interval
π_1 (Arsenal)	0.876	0.41	(0.342, 1.857)
π_2 (Bournemouth)	0.194	0.10	(0.066, 0.444)
π_3 (Middlesbrough)	1.233	0.56	(0.471, 2.585)
π_4 (Burnley)	1.528	0.74	(0.555, 3.330)
π_5 (Chelsea)	0.753	0.36	(0.280, 1.648)
π_6 (Crystal Palace)	1.125	0.51	(0.424, 2.407)
π_7 (Everton)	1.414	0.67	(0.528, 3.237)
π_8 (Hull)	3.188	1.58	(1.170, 7.486)
π_9 (Liverpool)	1.444	0.70	(0.567, 3.232)
π_{10} (Leicester)	3.839	1.95	(1.428, 8.767)
π_{11} (Man City)	0.465	0.22	(0.172, 1.029)
π_{12} (Man United)	0.095	0.05	(0.030, 0.228)
π_{13} (Sunderland)	2.765	1.34	(1.014, 6.078)
π_{14} (Southampton)	1.041	0.49	(0.400, 2.351)
π_{15} (Stoke)	1.533	0.73	(0.581, 3.370)
π_{16} (Swansea)	1.525	0.72	(0.582, 3.372)
π_{17} (Tottenham)	1.139	0.56	(0.424, 2.528)
π_{18} (Watord)	1.795	0.87	(0.670, 4.102)
π_{19} (West Brom)	1.216	0.58	(0.452, 2.705)
π_{20} (West Ham)	1.563	0.74	(0.575, 3.445)
ν	1.465	0.16	(1.172, 1.796)

Table 15: Posterior means and standard errors from the simulated data

Parameter	True value	MLE	Posterior mean (normalised)
π_1 (Arsenal)	0.027	0.027	0.030
π_2 (Bournemouth)	0.003	0.004	0.007
π_3 (Middlesbrough)	0.025	0.039	0.043
π_4 (Burnley)	0.026	0.052	0.053
π_5 (Chelsea)	0.054	0.022	0.026
π_6 (Crystal Palace)	0.016	0.036	0.039
π_7 (Everton)	0.048	0.047	0.049
π_8 (Hull)	0.045	0.127	0.111
π_9 (Liverpool)	0.082	0.047	0.050
π_{10} (Leicester)	0.121	0.157	0.134
π_{11} (Man City)	0.014	0.012	0.016
π_{12} (Man United)	0.001	0.001	0.003
π_{13} (Sunderland)	0.108	0.103	0.096
π_{14} (Southampton)	0.037	0.032	0.036
π_{15} (Stoke)	0.060	0.052	0.053
π_{16} (Swansea)	0.061	0.052	0.053
π_{17} (Tottenham)	0.049	0.036	0.040
π_{18} (Watord)	0.119	0.063	0.062
π_{19} (West Brom)	0.044	0.039	0.042
π_{20} (West Ham)	0.060	0.052	0.054
ν	1.450	1.523	1.465

Table 16: MLEs vs normalised posterior means

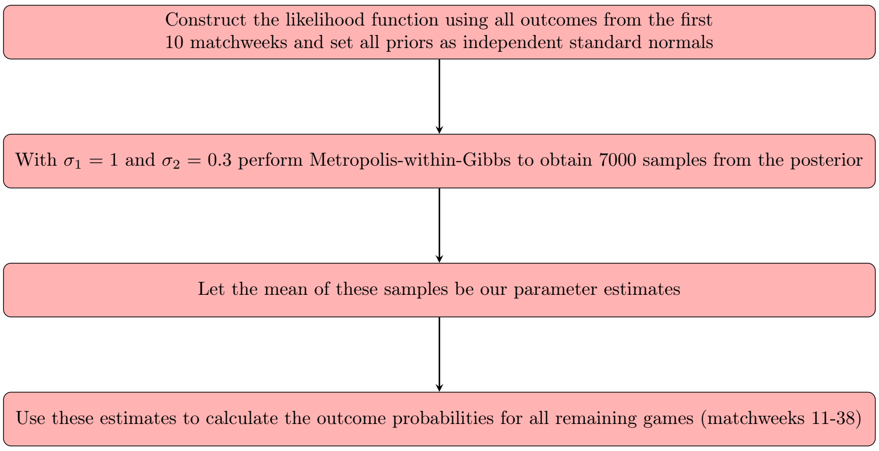
4.6 Predictions

4.6.1 Static model

As in the frequentist section, we will try to predict the outcomes of all the games from the 16/17 season, from matchweek 11 onwards. The procedure here will naturally be the same as the static model in the frequentist section apart from how we infer the parameters. Note that we will consider a chain to be sufficient if it is equivalent to at least 1000 i.i.d samples. Noticing that the autocorrelations never seem to exceed 0.73 and using our approximation of the effective sample size we get that

$$n \approx n_{eff} \frac{1+\rho}{1-\rho} \approx 1000 \left(\frac{1+0.73}{1-0.73} \right) \approx 6400$$

should be a sufficient number of iterations to the run chain and we will err on the side of caution and take $n = 7000$. Then the procedure is


Construct the likelihood function using all outcomes from the first 10 matchweeks and set all priors as independent standard normals
With $\sigma_1 = 1$ and $\sigma_2 = 0.3$ perform Metropolis-within-Gibbs to obtain 7000 samples from the posterior
Let the mean of these samples be our parameter estimates
Use these estimates to calculate the outcome probabilities for all remaining games (matchweeks 11-38)

Then taking the outcome with the highest predicted probability as the predicted outcome this achieves a classification accuracy of 55% and a mean RPS of 0.203.

4.6.2 Dynamic model

As in the frequentist section we now perform the same procedure from the static model but we update our estimates after each matchweek. The results for this model give a classification accuracy of 54.6% and a mean RPS of 0.202.

In the frequentist section we looked at how applying appropriate weights to matches based on how long ago they were played. This was possible since finding the MLE isn't very computationally expensive and so testing many different values of t_0 and w (our weight parameters) could be done relatively quickly. In this case however, running the Metropolis-within-Gibbs algorithm is much more costly and hence testing out many different values of t_0 and w isn't as feasible. This isn't overly disappointing since the predictions made by the two different methods of inference have been very similar so far. We therefore don't have much reason to think that inferring the parameters of the time-weighted model in a Bayesian sense will lead to significantly different predictions than what we obtained in the frequentist section.

5 Explanatory variables

5.1 Linear predictor

We have seen how we can build a model in terms of the log strengths, λ_i, of all the teams and an extra parameter controlling the prevalence of draws, ν. In this section we will try to improve the predictive power of our model by assuming that the log strengths are of the form

$$\lambda_i = \alpha + \sum_{k=1}^{p} \beta_k f_{ik} = \alpha + \beta^T f_i$$

where $\beta = (\beta_1, ..., \beta_p)$ is our new parameter vector, $f_i = (f_{i1}, ..., f_{ip})$ is a vector of features related to team i and α is the intercept term. Previously we assigned probabilities to outcomes according to

$$P(i > j) = \frac{e^{\lambda_i}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}} = \frac{1}{1 + e^{\lambda_j-\lambda_i} + \nu\sqrt{e^{\lambda_j-\lambda_i}}}$$

$$P(i \leftrightarrow j) = \frac{\nu\sqrt{e^{\lambda_i+\lambda_j}}}{e^{\lambda_i} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\lambda_j}}} = \frac{\nu}{\sqrt{e^{\lambda_i-\lambda_j}} + \sqrt{e^{\lambda_j-\lambda_i}} + \nu}.$$

Now we can substitute in the for the log strengths the linear combinations of explanatory variables to get (notice the α terms will cancel)

$$P(i > j) = \frac{1}{1 + e^{\beta^T f_j - \beta^T f_i} + \nu\sqrt{e^{\beta^T f_j - \beta^T f_i}}} = \frac{1}{1 + e^{\beta^T (f_j - f_i)} + \nu\sqrt{e^{\beta^T (f_j - f_i)}}}$$

$$P(i \leftrightarrow j) = \frac{\nu}{\sqrt{e^{\beta^T f_i - \beta^T f_j}} + \sqrt{e^{\beta^T f_j - \beta^T f_i}} + \nu} = \frac{\nu}{\sqrt{e^{\beta^T (f_i - f_j)}} + \sqrt{e^{\beta^T (f_j - f_i)}} + \nu}.$$

We can then perform inference in much the same way as before to find the parameters β and ν, and use our estimates to predict future games. Of course before we can do this we need to specify what explanatory variables we will use.

5.2 Home advantage

Arguably the most important explanatory variable is that of the location of the game. In particular, a team that is playing at home, at their own stadium, is more likely to win compared to if that same team was playing away i.e. a team gains a slight advantage if they are playing at home. To confirm that this is the case we take a look at the data from the 16/17 Premier League season. We show for 6 different teams how their performances differ when at home and when playing away (figures 23 and 24).

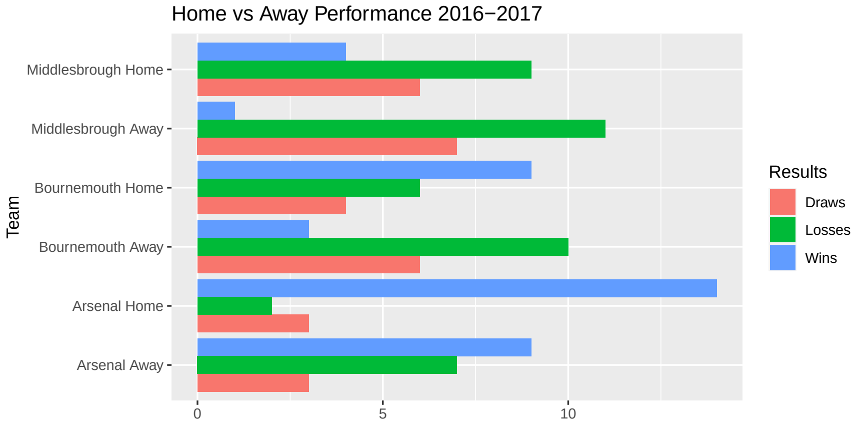


Figure 23: Home vs away (1)

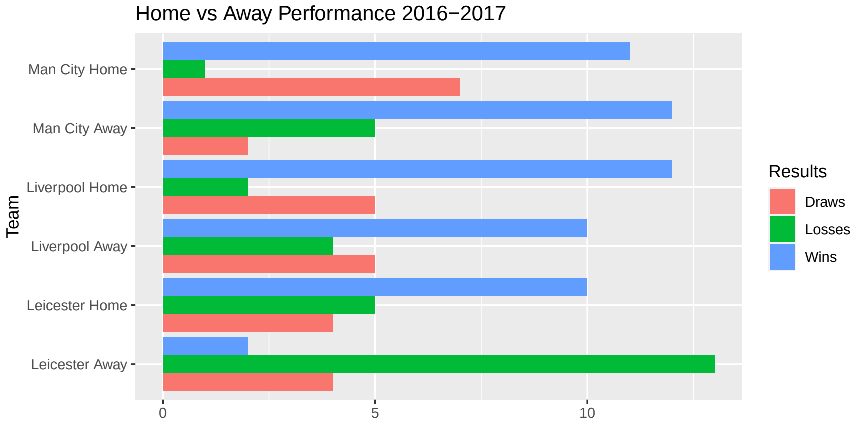


Figure 24: Home vs away (2)

Looking at the blue bars it is clear that teams win more when they are playing at home. All teams conform to this except for Man City who won one more game away than at home. In football however, a team gets one point for a draw and three points for a win and so Man City got 40 points at home and 38 points away. Hence Man City did still perform better at home on the whole. In fact, all teams in this season got more points at home than they did away apart from Man United and Crystal Palace who both got just one more point in their away games.

We want to confirm mathematically what we can see in the plots. To do this we will investigate whether the average number of wins achieved by a home team is significantly greater than the average number of wins achieved by an away team. We have

$$x_1 = (14, 9, 4, 10, 17, 6, 13, 8, 12, 10, 11, 8, 3, 6, 7, 8, 17, 8, 9, 7),$$
$$x_2 = (9, 3, 1, 1, 13, 6, 4, 1, 10, 2, 12, 10, 3, 6, 4, 4, 9, 3, 3, 5),$$

where x_1 is our observed sample for the number of home wins and similarly x_2 is our observed sample for the number of away wins. These samples are just the number of home and away wins for each of the 20 teams. We plot the distributions of the different samples

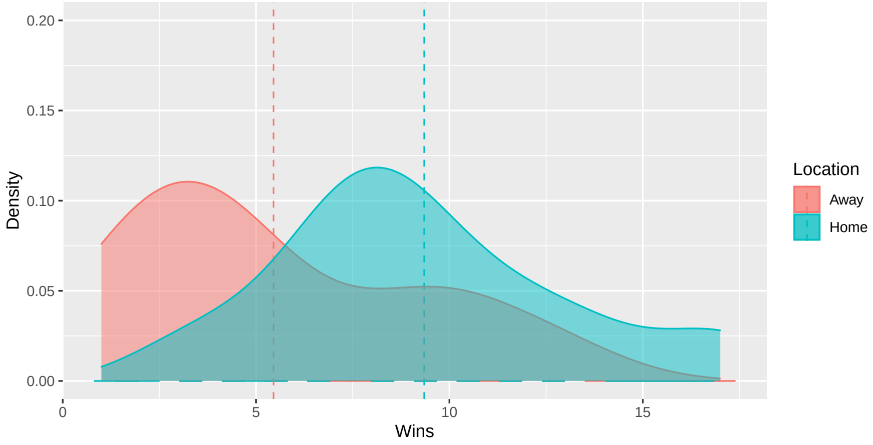


Figure 25: Distributions of home wins and away wins

with the dashed lines to show the sample means $\bar{x}_1 = 9.35$ and $\bar{x}_2 = 5.45$. We want to test the null hypothesis that these two samples have come from the same distribution, and to do this we will use a non-parametric test known as the permutation test. First we calculate the observed test statistic $T_{obs} = \bar{x}_1 - \bar{x}_2 = 3.9$ and we want to find the probability of observing a test statistic at least as extreme as 3.9. To do this notice that if the null hypothesis is true and the samples are both from the same distribution, then we can pool the samples together and treat all 40 observations as one sample from some distribution. Now to find the distribution of the test statistic we perform n times the following procedure,

1. Randomly permute the 40 observations and assign the first 20 values to a group x_1^* and assign the remaining values to another group x_2^*.
2. Calculate the difference in the means, $T = \bar{x}_1^* - \bar{x}_2^*$, and store this value.

When performing a permutation test one finds the distribution of the test statistic by calculating its value under every possible permutation of the pooled sample. However in this case the total number of permutations is $40! \approx 8 \times 10^{47}$, hence we approximate the distribution of the test statistic by performing our above procedure $n = 10^6$ times. Plotting the distribution of our 10^6 different values for T gives

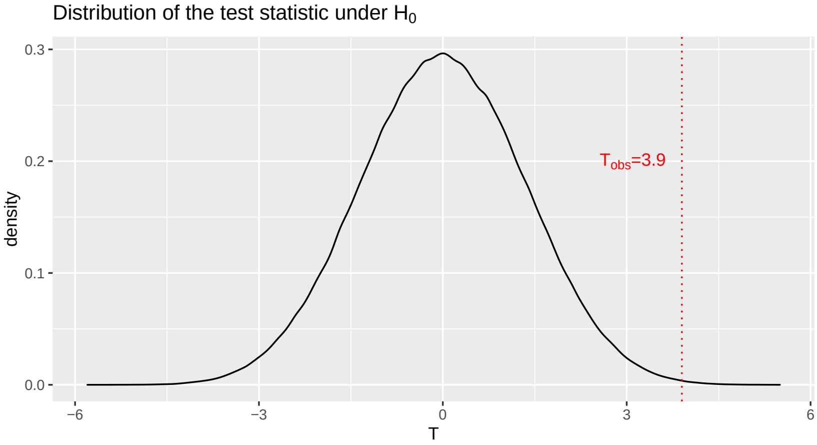


Figure 26: Approximate distribution of the test statistic

We can see that under the null hypothesis our observed test statistic is quite an extreme value. Our Monte Carlo estimate of the p-value is given by the proportion of our 10^6 different values of T which are greater than or equal to 3.9, which we calculate to be 0.0014. This tells us that it is unlikely that we would see a test statistic this extreme under the null hypothesis and we therefore have substantial evidence to reject the null hypothesis and we conclude that the number of games won by the home team is significantly greater than the number won by the away team. Hence our first explanatory variable will be a categorical variable specifying if the team is playing at home or away. More specifically this explanatory variable is equal to 1 if the team is playing at home and 0 otherwise.

5.3 Team form

Our next explanatory variable will be related to the form of the team. In football, it is common for teams to go through periods of good or bad form. To see an example of this we visualise the results of Crystal Palace from the 16/17 season in figure 27.

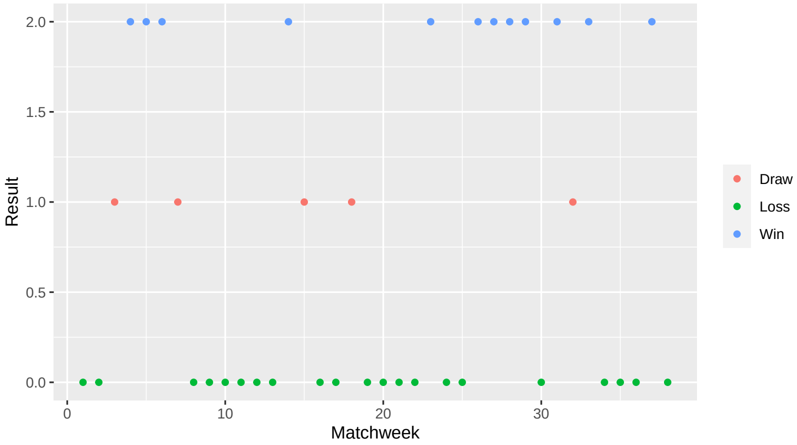


Figure 27: Crystal Palace form, 16/17

Looking at the results between matchweeks 8 and 25 we can see that Crystal Palace lost most of their games i.e. they were on a run of bad form. However later on in the season it can be seen that they start performing better and seem to have a small run of good form in matchweeks 26 to 33. It therefore seems more likely that a team will win if it is in a run of good form and so we define our next explanatory variable to be equal to the average number of points won by the team in its most recent 4 matches. Note that at the start of the season when a team hasn't yet played 4 matches, the team's form is defined to be the average number of points won in all of the teams previous games from the current season. Additionally for the first week where we have not yet got any observations we just set each teams form to 1.

5.4 Goals scored/conceded

The next two variables we will consider is the number of goals scored and conceded by the team in its recent matches. The reasoning behind these variables is identical to that of the team form. The idea is that we can observe how many goals a team has been scoring or conceding in its recent matches and hopefully from this have an idea of how well it might perform in its next match. Firstly, we want to confirm that a stronger team does in fact score more goals. To do this we will use the final league standings from the 16/17 season and split the 20 teams into 3 different groups: the teams that finished in the top 5, the middle 10 teams and the bottom 5 teams. The idea is that we want to test whether the average number of goals scored by a team in the top 5 is greater than the average number of goals scored by a team in the middle 10, and so on. We proceed by showing the distributions of the samples we have for the 3 groups in figure 28.

The dashed lines are the means for each of the groups. It seems clear from the plot that the higher ranked teams score more goals on average but

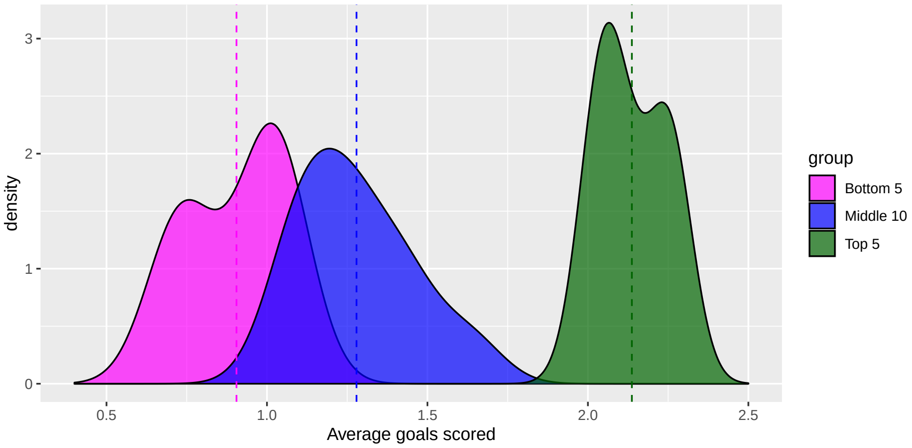


Figure 28: Average goals scored by group

we wish to investigate this mathematically. A useful test of whether our multiple population means are equal is the analysis of variance (ANOVA). To make sure that it is appropriate to perform an ANOVA test we must first check that our 3 different samples are normally distributed and also that the variance is the same for the 3 samples. To check the normality of the samples we plot each of their distributions next to their approximate normal distribution, that is, the normal distribution with mean and variance equal to the sample mean and sample variance (figures 29, 30 and 31).

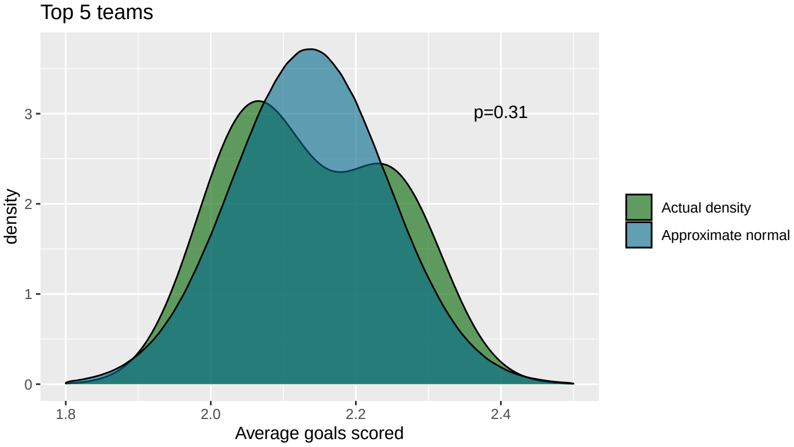


Figure 29: Distribution of top 5 group vs approximate normal

Also annotated on each of the plots is the p-value from performing a Shapiro-Wilks test on the sample. This is the most powerful known test of normality

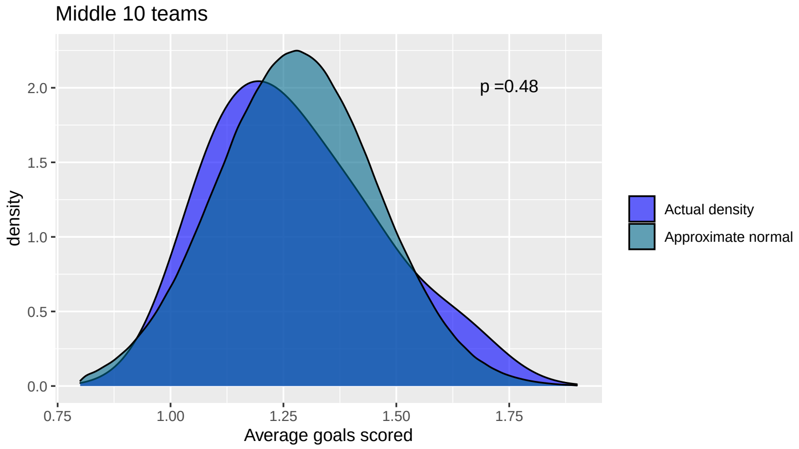


Figure 30: Distribution of middle 10 group vs approximate normal

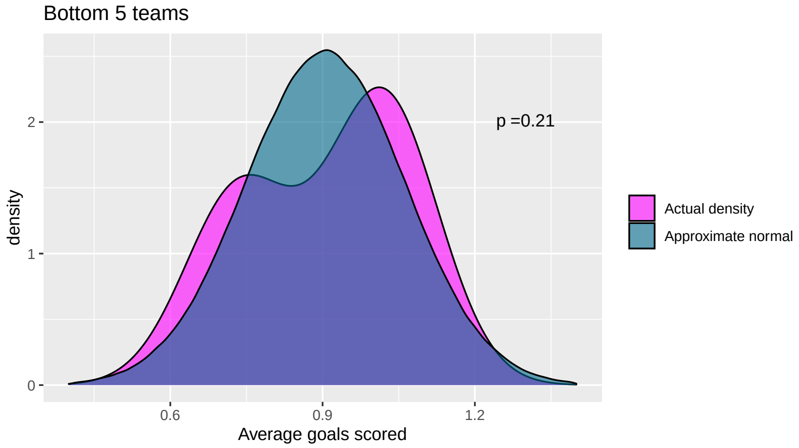


Figure 31: Distribution of bottom 5 group vs approximate normal

and it tests the null hypothesis that a sample is from a normal distribution. As we can see none of the p-values are significantly small and hence we have no reason to reject that our 3 samples are from normal distributions.

It remains to show that the populations have the same variance, which we can show by performing an F-test for the equality of variances. This tests the null hypothesis that two normal populations have equal variance. An F-test between the 'top 5' sample with the 'middle 10' sample gives a p-value of 0.34, comparing the 'top 5' with the 'bottom 5' gives a p-value of 0.48 and comparing the 'middle 10' with the 'bottom 5' yields a p-value of 0.87. Again, none of these p-values are small and so it is reasonable for us to assume that our 3 populations have similar variances.

We have seen no evidence that we can't assume normality and equal variances of our populations, hence it is appropriate for us to perform an ANOVA to compare the means of the 3 populations. Running an ANOVA returns a p-value of roughly 2×10^{-9} hence we can have a lot of confidence that there is a difference in the population means. The problem with this test however is that it isn't able to tell us which of the groups were significantly different, just that at least two of the groups were. To investigate further the differences between the different groups we can perform a post hoc test known as Tukey's HSD (honestly significant difference) test. This test reports a p-value for every pairwise comparison of means and in this case it gives 3 p-values of $0.001, 2 \times 10^{-9}$ and 6×10^{-8}. We won't worry about which p-value is associated with which comparison of means since they are all very significant anyway. P-values this small strongly indicate that all population means are significantly different from each other, and we conclude that teams that ranked higher did score more goals.

To avoid repeating ourselves we won't reproduce all the same plots and p-values that are produced by an identical analysis of the average number of goals conceded by the 3 groups. In summary, all the same assumptions are still met and an ANOVA gives a p-value of 0.03. Tukey's HSD test then yields p-values of 0.02 for the comparison of the top 5 and middle 10 groups, 0.003 for the comparison of top 5 and bottom 5 and 0.30 for the comparison of middle 10 and bottom 5. In this case there is some evidence that the higher ranked teams conceded less goals, although it isn't as clear as it was previously and it seems there is no significant difference between the average number of goals conceded between the middle 10 and bottom 5 teams. To confirm visually what these p-values tell us we show the distributions of the 3 samples

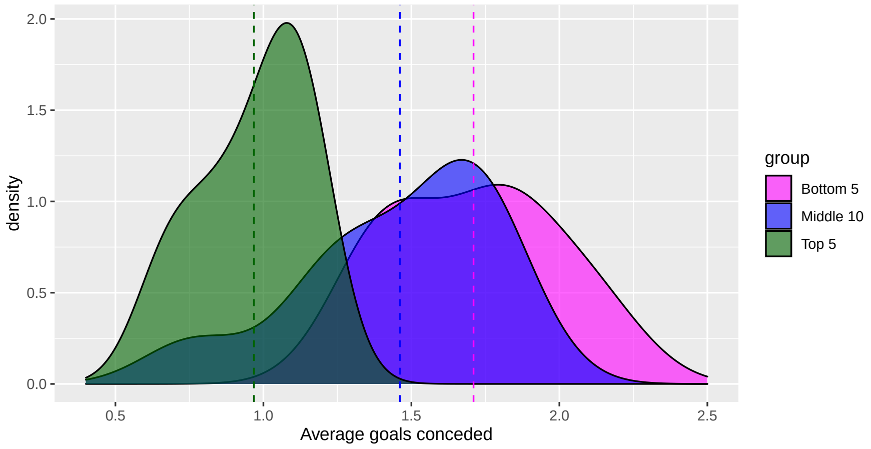


Figure 32: Average goals conceded by group

We can see that the distributions for the middle 10 and bottom 5 overlap quite a lot, explaining why this pairwise comparison had a non-significant p-value.

It seems that in general the higher a team is ranked, the more goals it scores and the less goals it concedes. We therefore define our final two explanatory variables to be the average number of goals scored by the team in its five most recent matches and the average number of goals conceded by the team in its five most recent matches. For matchweeks 2-5 when we don't yet have five observations the explanatory variables will be equal to the average number of goals scored/conceded in all previous matches from the current season. Finally in the first week when we have no observations we will set both explanatory variables equal to one for all teams.

5.5 Predictions

We will now implement the model outlined in section 5.1 where we predict the strengths of the teams through a linear combination of features. From now on we will always estimate parameters by maximum likelihood estimation. We do this since we want to evaluate how different combinations of our explanatory variables perform, and MLE is much quicker than running the Metropolis-within-Gibbs algorithm. Also we will only consider the 'dynamic' setting here where we update our parameter estimates every week. We compare how using different combinations of the explanatory variables affects the classification accuracy:

Explanatory variables	Classification (%)
Home/away, form, goals scored, goals conceded	56.4
Home/away, form, goals scored	57.1
Home/away, form, goals conceded	55.7
Home/away, goals scored, goals conceded	57.9
Goals scored, form, goals conceded	51.1
Home/away, form	55.7
Home/away, goals scored	57.5
Home/away, goals conceded	53.6
Goals scored, goals conceded	51.4
Goals scored, form	52.1
Goals conceded, form	51.4

Table 17: How the choices of explanatory variables affects the classification

Looking at these results it seems as though the only significant feature is that of the home advantage, since all classifications related to models without the home/away variable are lower. This makes sense since the home advantage is a well established effect whereas it isn't as clear that the other variables would have much predictive power. This motivates us to make a model that includes only the home advantage as an explanatory variable. To do this we go back to our model which had the teams' log strengths as the parameters, but we add in a new parameter α in an attempt to incorporate the home advantage effect. In particular, if team i is the home team playing against team j, we have

$$P(i > j) = \frac{e^{\lambda_i + \alpha}}{e^{\lambda_i+\alpha} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\alpha+\lambda_j}}}$$

$$P(i \leftrightarrow j) = \frac{\nu\sqrt{e^{\lambda_i+\alpha+\lambda_j}}}{e^{\lambda_i+\alpha} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\alpha+\lambda_j}}}$$

$$P(j > i) = \frac{e^{\lambda_j}}{e^{\lambda_i+\alpha} + e^{\lambda_j} + \nu\sqrt{e^{\lambda_i+\alpha+\lambda_j}}}.$$

We add the term $\alpha > 0$ to the log strength of the home team, therefore hoping to capture the fact that teams are generally stronger at home. Implementing this model we find that our predictions successfully classify 60.7% of the matches and we also obtain a mean rank probability score of 0.189, which is the best performance of any of the models we have seen so far.

6 Conclusion

Given that the home advantage model was the best in terms of predictive performance it remains to test the model on some new unseen data. We implement the home advantage model in our dynamic setting on the 2017/2018 Premier League season and attempt to predict the outcomes from matchweek 11 until the end of the season. On this new data we obtain a classification accuracy of 53.2% and a mean RPS of 0.198. This classification accuracy is noticeably lower than the one we obtained previously (60.7%) which at first glance makes us wonder whether we over-fitted the model to the data from the 16/17 season, leading to a model that doesn't generalise well and hence performs poorly on new data sets. However, remember that we want our models to have a classification accuracy higher than the simple model which always predicts a home win. So for the 16/17 season predictions where we obtained a classification accuracy of 60.7%, we have that a model always predicting a home win would have a classification accuracy of 51.8%, and so the home advantage model is around 9% higher than the null classification. Whereas the null classification for the matches we tried to predict from the 17/18 season is only 45.7%, meaning the classification accuracy of the home advantage model in this case (53.2%) is around 8% higher than the null classification. Therefore it seems that using the home advantage model on the new data is actually of comparable performance and that the home advantage model does generalise well.

We obtained predictive accuracies of 60.7% and 53.2% for the 16/17 and 17/18 seasons respectively, when using maximum likelihood estimation as our method of inference. Attempting the same predictions but taking the posterior means as our estimates, we obtain predictive accuracies of 61.8% and 53.6%. In both cases the Bayesian approach proves superior, but only by a small margin. In general the two methods of inference seemed to give us very similar levels of accuracy. The most notable difference between the methods was their computational demands. Maximum likelihood estimation proved far quicker than finding the posterior means. As a result of this, when investigating the effects of a time weighting component in section three we weren't able to consider the Bayesian approach since it was too computationally expensive. We encountered the same problem in section five when we looked at using every different combination of explanatory variables. Again, the Bayesian approach wasn't quick enough to allow us to look at all the different models, hence we had to rely on maximum likelihood estimation. Therefore, given that the predictive capabilities were almost the same, it seems that maximum likelihood estimation has been the better method for us. On the other hand, it is the Bayesian approach which shows more promise. Recall that Constantinou et al. (2012) were able to create a profitable betting strategy by combining the subjective opinion of an expert and the objective forecasts of their Bayesian network. If we were to use a similar strategy, it is clear that the easiest way for us to incorporate the subjective opinion is through the specification of our prior distributions. Hence it seems as though we are missing out by setting the relatively uninformative priors that we did.

A Code

The dataset and all code used in this paper is available at https://github.com/mattbarrett98/Bradley-Terry-project.

References

A. Agresti. *Categorical data analysis*, volume 482. John Wiley & Sons, 2003.

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444. URL http://www.jstor.org/stable/2334029.

S. R. Clarke and J. M. Norman. Home ground advantage of individual clubs in English soccer. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 44(4):509–521, 1995. ISSN 00390526, 14679884. URL http://www.jstor.org/stable/2348899.

A. Constantinou and N. Fenton. Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models. *Journal of Quantitative Analysis in Sports*, 8, 01 2012. doi: 10.1515/1559-0410.1418.

A. Constantinou, N. Fenton, and M. Neil. pi-football: A Bayesian network model for forecasting association football match outcomes. knowledge-based systems, 36, 322-339. *Knowledge-Based Systems*, 36:332–339, 12 2012. doi: 10.1016/j.knosys.2012.07.008.

R. R. Davidson. On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970. ISSN 01621459. URL http://www.jstor.org/stable/2283595.

M. J. Dixon and S. G. Coles. Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(2):265–280, 1997. doi: https://doi.org/10.1111/1467-9876.00065. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9876.00065.

C. J. Geyer. Introduction to MCMC, n.d. URL https://si.biostat.washington.edu/sites/default/files/modules/Geyer-Introduction%20to%20markov%20chain%20Monte%20Carlo_0.pdf.

D. Henderson and L. Kirrane. A comparison of truncated and time-weighted Plackett–Luce models for probabilistic forecasting of Formula One results. *Bayesian Analysis*, 13, 02 2017. doi: 10.1214/17-BA1048.

M. D. Hoffman and A. Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. URL http://jmlr.org/papers/v15/hoffman14a.html.

D. R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384–406, 2004. ISSN 00905364. URL http://www.jstor.org/stable/3448514.

G. L. Jones. On the Markov chain central limit theorem. *Probability Surveys*, 1(none):299 – 320, 2004. doi: 10.1214/154957804100000051. URL https://doi.org/10.1214/154957804100000051.

J.-W. Kuo, P.-J. Cheng, and H.-M. Wang. Learning to rank from bayesian decision inference. pages 827–836, 01 2009. doi: 10.1145/1645953.1646058.

S. Lock. Key data on the global sports betting sector 2020, 2021. URL https://www.statista.com/statistics/1154681/key-data-global-sports-betting-industry/.

R. D. Luce. *Individual Choice Behavior: A Theoretical analysis*. Wiley, New York, NY, USA, 1959.

J. Lukas. Bradley-Terry-Luce scale of taste qualities of champagne. *Zeitschrift für experimentelle und angewandte Psychologie*, 38:605–19, 02 1991.

M. J. Maher. Modelling association football scores. *Statistica Neerlandica*, 36(3):109–118, 1982. doi: https://doi.org/10.1111/j.1467-9574.1982.tb00782.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9574.1982.tb00782.x.

J. N. S. Matthews and K. P. Morris. An application of Bradley-Terry-type models to the measurement of pain. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44(2):243–255, 1995. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2986348.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2346567.

P. V. Rao and L. L. Kupper. Ties in paired-comparison experiments: A generalization of the Bradley-Terry model. *Journal of the American Statistical Association*, 62(317):194–204, 1967. doi: 10.1080/01621459.1967.10482901.

C. Reep and B. Benjamin. Skill and chance in association football. *Journal of the Royal Statistical Society. Series A (General)*, 131(4):581–585, 1968. ISSN 00359238. URL http://www.jstor.org/stable/2343726.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997. ISSN 10505164. URL http://www.jstor.org/stable/2245134.

D. Selby and D. Firth. PageRank and the Bradley-Terry model, n.d. URL https://warwick.ac.uk/fac/sci/statistics/staff/research_students/selby/pagerank_poster.pdf.

Stan Development Team. RStan: the R interface to Stan, 2020. URL http://mc-stan.org/. R package version 2.21.2.

M. Tilp and S. Thaller. Covid-19 has turned home advantage into home disadvantage in the German soccer Bundesliga. *Frontiers in Sports and Active Living*, 2:165, 2020. ISSN 2624-9367. doi: 10.3389/fspor.2020.593499. URL https://www.frontiersin.org/article/10.3389/fspor.2020.593499.

A. Tsokos, S. Narayanan, I. Kosmidis, G. Baio, M. Cucuringu, G. Whitaker, and F. Király. Modeling outcomes of soccer matches. *Machine Learning*, 108, 01 2019. doi: 10.1007/s10994-018-5741-1.

H. Turner and D. Firth. Bradley-Terry models in R: The BradleyTerry2 package. *Journal of statistical software*, 48, 05 2012. doi: 10.18637/jss.v048.i09.

S. Wood. *Generalized Additive Models: An Introduction With R*, volume 66. 01 2006. ISBN 9781315370279. doi: 10.1201/9781315370279.

E. Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Math Z*, 29:436–460, 1929. doi: 10.1007/BF01180541.