Stockholms universitet

Bayesian Logistic Regression

Mikael Rizvanovic

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2023:4
Matematisk statistik
Januari 2023

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Bachelor Thesis **2023:4**
http://www.math.su.se

Bayesian Logistic Regression

Mikael Rizvanovic*

June 2023

Abstract

In this thesis we introduce Bayesian statistics and a Bayesian non-hierarchical model for logistic regression. We also discuss how this model can be generalized into a hierarchical regression model. For both models we employ the Metropolis-Hastings algorithm, which is a Markov Chain Monte Carlo method that can be used to train Bayesian models. We then apply this theory to a real data set consisting of professional tennis matches and evaluate the result. Special emphasis is given to a discussions about model building and model testing in a Bayesian setting.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: mikael.rizvanovic@gmail.com. Supervisor: Ola Hössjer & Kristoffer Lindensjö.

Acknowledgements

This is a bachelor thesis of 15 ECTS in Mathematical Statistics at the Department of Mathematics at Stockholm University. I am grateful for the guidance from my supervisors Ola G. H. Hössjer and Kristofer Lindensjö. Their valuable help has facilitated this work.

Contents

1 Introduction

Bayesian statistics is one of many possible approaches to applying probability theory to statistical problems. The perhaps most popular approach is frequentist statistics, where probabilities are interpreted as long-run frequencies of random events in repeated trails with no prior beliefs or information. Bayesian statistics provides us with a mathematical framework to incorporate prior information in a *prior distribution* and then update our beliefs based on data. This is done using a result called Bayes theorem and it gives us access to a full *posterior distribution* from which to make inference.

The basis for Bayesian statistics was first introduced by reverend Thomas Bayes in 1763. Not much would happen for the next 50 years until Pierre-Simon Laplace independently rediscovered Bayes theorem in the early 1800s. Laplace would make several important contributions to the theory and through his work Bayesian statistics would become the dominant statistical approach until the early 1900s when the frequentist view rapidly took over [14]. A few early contributors to the frequentist approach were Karl Pearson, Ronald Fisher and Jerzy Neyman. They, and others, managed to derive a framework for statistical inference that is powerful enough to solve many problems that were not possible to solve using Bayesian statistics at that time [14]. The reason Bayesian statistics were not able to solve those problems was because it is typically not possible to derive the posterior distribution in closed form. Therefore, outside of a few toy examples, posterior inference is typically done with the help of numerical methods.

One family of such methods is known as Markov Chain Monte Carlo (MCMC) simulation, the first of which was introduced by Metropolis et al. in 1953 [15]. The discovery that simulations obtained from this method could be used for simulating draws from the posterior gave rise to a flurry of new research in the field. In a fairly short period of time many improvements were made, both in the effectiveness of the sampling schemes and in Bayesian theory more generally [8, chapter 2]. However, Bayesian statistics was still just a niche. Besides a few notable exceptions it was seldom used to solve real problems. This is because as we start to work with more complex distributions and problems the number of simulations needed before we have an adequate description of the posterior distribution becomes a constraint. But as computational power increased, became cheaper and new more efficient simulation methods were discovered, the adaptation of Bayesian statistics started to take root in several disciplines and they have become especially entrenched in a few, such as ecology and genetics. Bayesian methods have also seen increased use in solving specific problems, often in combination with non Bayesian methods. Two such examples are filtering problems and generative modeling [8].

In this thesis we will give a broad introduction to Bayesian statistics and model building. We will start by introducing logistic regression, which is a popular

model for predicting binary comes. We will then introduce Bayesian statistics and the Bayesian equivalence of logistic regression followed by a discussion about the Metropolis-Hastings algorithm, which is a popular MCMC method, and a couple of methods for assessing whether or not the simulations made accurately represent the posterior distribution. Finally, we will build and analyze a Bayesian logistic regression (BLR) model for predicting the outcomes on tennis matches.

2 Theory

2.1 Logistic Regression

All theory in this section 2.1 is taken from [2, chapter 5], unless otherwise specified.

Logistic regression is a popular model used in a wide range of situations to predict binary outcomes, such as sick/healthy, pass/fail, win/lose and so on. Mathematically, the model is defined in the following way.

Let Y be a binary variable and $\boldsymbol{X} = (X_1, X_2, ..., X_p)^T$ a vector containing p explanatory variables. Suppose we have a sample $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ of size n, with $y_i \in \{0, 1\}$ the binary outcome and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ containing the p explanatory variables of observation i. Then then the win probability

$$\pi(\boldsymbol{x}_i) = P\left(Y = 1 | \boldsymbol{X} = \boldsymbol{x}_i\right) = 1 - P\left(Y = 0 | \boldsymbol{X} = \boldsymbol{x}\right)$$

of observation i can be evaluated with the (multiple) *logistic regression model*

$$\pi(\boldsymbol{x}_i) = \frac{\exp\left(\alpha + \sum_{j=1}^p \beta_i x_{i,j}\right)}{1 + \exp\left(\alpha + \sum_{i=1}^p \beta_i x_{i,j}\right)}, \tag{2.1}$$

where α is the model intercept and $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)$ contains the effect parameters.

For practical purposes it is often easier to work with the *log odds* transformation, a.k.a. the logit transformation

$$\text{logit}\left[\pi(\boldsymbol{x}_i)\right] = \log \frac{\pi(\boldsymbol{x}_i)}{1 - \pi(\boldsymbol{x}_i)} = \alpha + \beta_1 x_{i,1} + \beta_2 x_{i,2} + ... + \beta_p x_{i,p} \tag{2.2}$$

These parameter values are unknown to us beforehand and must be estimated somehow. In the frequentist setting this is done by maximizing the *likelihood function*.

Let $f(y_i | \boldsymbol{x}_i, \alpha, \boldsymbol{\beta}) = \pi(\boldsymbol{x}_i)^{y_i}(1 - \pi(\boldsymbol{x}_i)^{1-y_i}$ refer to the probability of observation

i. The likelihood function is the joint probability distribution of observed data, i.e.

$$\mathcal{L}(\alpha, \boldsymbol{\beta}) = \prod_{i=1}^{n} f(y_i|\boldsymbol{x}_i, \alpha, \beta)$$

and it tells us how likely our observed data are to happen given some parameter values. The maximum likelihood estimate contains the values for α and $\boldsymbol{\beta}$ that maximize the probability of observing our data. How this maximum is found is not relevant to this thesis but the interested reader can see [2, section 5.5.1] for logistic regression specifically or [3, section 2.1.1] for MLE more generally.

2.2 Bayesian Statistics

In Bayesian inference the parameter, θ, we want to study is random with a *prior* distribution $f(\theta)$. After observing data, $X = x$, the *posterior* distribution $f(\theta|x)$ can be computed using *Bayes rule*, which is derived by using the property of conditional probabilities twice [1]. The conditional probability for two random variables x and θ can be written as

$$f(x|\theta) = \frac{f(x, \theta)}{f(\theta)}, \tag{2.3}$$

This means that we can write the joint probability of x and θ as

$$\begin{aligned} f(x, \theta) &= f(x|\theta)f(\theta) \\ &= f(\theta|x)f(x). \end{aligned}$$

Using this equation we can easily derive the formula for the posterior

$$f(\theta|x) = \frac{f(\theta)f(x|\theta)}{f(x)} = \frac{f(\theta)f(x|\theta)}{\int_\Theta f(\theta')f(x|\theta')d\theta'}, \tag{2.4}$$

where θ is an unknown parameter of interest that belongs to the parameter space Θ and x is the data. We used $f(x) = \int f(\theta, x)d\theta = \int f(\theta)f(x|\theta)d\theta$ in the second denominator, which follows from the law of total probability. This is how the posterior distribution is calculated.

We will now explain what every term in equation (2.4) corresponds to.

- $f(\theta)$ is our prior distribution, this is what we believe the distribution is before taking any evidence (data) into account. We will discuss possible prior choices in section 2.2.1.
- $f(x|\theta)$ is the likelihood function, as briefly discussed in section 2.1. This tells us what the probability of observing x is given θ.
- $f(\theta|x)$ is the posterior distribution. This can be seen as our updated belief after we have seen the data, x.

- $f(x)$ is the evidence. This is a constant, integrating (or summing) over all possible values of θ. We can think of this as a normalizing constant which makes the posterior integrate to 1. This constant can be very hard or impossible to compute analytically. Instead numerical methods or approximations are generally used, as will be discussed in section 2.3.

In some applications we can use proportionality arguments to simplify calculations [3]

$$f(\theta|x) \propto f(\theta)f(x|\theta) \tag{2.5}$$

This is no longer a well defined density function (it does not integrate to 1) but it is proportional to the posterior, meaning that the two equations differ by a constant multiplier. Therefore, some estimates will be the same, like the mean, the median, credibility intervals etc.

One of the goals of Bayesian statistics is to provide us with a mathematically sound procedure to incorporate prior beliefs into our model so that we together with any evidence at hand can produce an updated posterior belief. This allows us to incorporate our own opinions in a way that is hard to achieve with data alone. This posterior can then itself be used as a prior if we come across new data. Hence, by repeatedly applying Bayes rule we are able to continually adjust our beliefs under new data [1].

2.2.1 Choice of prior distribution

Bayesian inference allows us to incorporate our prior beliefs through a prior distribution. We will consider three broad categories of priors: Informative (or strong), non-informative and weakly informative [1, chapter 2]. How informative our prior is will determine how much evidence we need (i.e. strength of the likelihood) before we abandon our prior beliefs. A strong prior will require a lot of evidence (data) to overturn while a non-informative prior will be overturned immediately and have a minimal impact on the posterior. The rationale of the non-informative prior is that we will "let the data speak for themselves" in order for posterior inference to be unaffected by information external to the current data [1]. However, we usually have some prior beliefs. For example, if we flip a coin we expect the outcome to be heads or tails even if it is possible for the coin to land on its side/edge. A weakly informative prior contains enough information to regularize the posterior and thus keep it roughly within reason [1]. In this section we will discuss a few different prior distributions and how to choose a prior.

Conjugate priors is often a pragmatic approach for choosing a prior. That is, choosing a prior in such a way that the posterior belongs to the same family of distributions. More formally, let $\mathcal{F}$ be a class of sampling distributions $p(x|\theta)$ (i.e the distribution of the likelihood) and let $\mathcal{P}$ be a class of prior distributions for θ. If

$$f(\theta|x) \in \mathcal{P} \text{ for all } p(\cdot|\theta) \in \mathcal{F} \text{ and all } p(\cdot) \in \mathcal{P}$$

then we say that the class $\mathcal{P}$ is *conjugate* for $\mathcal{F}$ [1, section 2.4].

Conjugate priors are mathematically convenient to work with. They are easy to deal with computationally and can often be put into analytic form, making it easier to interpret the results. They often give good approximations but since the only influence the likelihood has on the posterior is in changing the parameter values they make a strong assumption about data beforehand, which is not always justifiable [1]. For this reason they are considered to be informative priors.

If we instead want to use an non-informative prior we can either choose a (roughly) symmetric distribution, like the Gaussian distribution or t-distribution, and set the scale to infinity (analytically, computationally to some high value), or we can choose a (locally) uniform prior. Then the prior will be proportional to a constant and the posterior will be proportional to the likelihood. This means that, if the parameter space is not bounded, we could end up with an improper prior (density does not integrate to 1). However, we might still get a proper posterior [1, section 2.8].

Another set of non-informative priors are *Jeffreys prior*, these have the added benefit of being invariant under one-to-one transformations of the parameter. Specifically, if we consider a one-to.one transformation $\phi = h(\theta)$, then expressing the same beliefs in ϕ as for the prior density $f_\theta(\theta)$ of θ is equivalent to

$$f_\phi(\phi) = f_\theta(\theta) \left| \frac{d\theta}{d\phi} \right| = f_\theta(\theta)|h'(\theta)|^{-1}.$$

From this it follows that the form of $f_\phi(\phi)$ is independent of ϕ only if $h(\cdot)$ is linear [3, section 6.3.3]. Jeffreys principle states that any choice of prior should be invariant under one-to-one transformations . From Jeffreys principle it follows that the non-informative prior is defined as $p(\theta) \propto [J(\theta)]^{1/2}$ where $J(\theta)$ is the Fisher information for θ [1], i.e:

$$J(\theta) = -E\left(\frac{d^2 \log p(Y|\theta)}{d\theta^2} \middle| \theta \right).$$

This method is known as Jeffreys prior (or sometimes Jeffreys rule).

Jeffreys prior can be used in multi-parameter models as well but the result is more controversial. Jeffreys prior in multi-parameter models will still fulfill the invariance property but we often get contradicting results depending on what information we have access to [3, section 6.5.2].

There are two ways to construct a *weakly informative* prior [1, section 2.9]

- Start with a noninformative prior and add information.
- Start with a strongly informative prior and remove information to account for uncertainty.

The purpose behind both methods is to get a prior that regularizes the result enough so that inferences are reasonable, but not more than that. What makes a prior weakly informative depends on the situation, the same prior could be non-informative in one situation and strong in another. This will depend on data, modeling objectives and model structure. For example, a uniform prior is usually the weakest possible prior if we model a large and normally distributed data set directly from Bayes rule (2.4) but might be very strong if we are trying to model the prevalence of a rare disease [1, section 2.9]. Furthermore, a reasonable prior on one scale could be unreasonable on another, for example on the log scale in logistic regression [1, section 16.3]. The prior's influence could also propagate in a model. If we assume prior independence between the parameters then many individually weakly informative priors could together become strongly informative [9]. If we use a multilevel (hierarchical) model then non-informative variance priors could propagate through levels and together give unreasonable variances [1, section 5.7]. We will touch upon this topic more in the upcoming sections, but deriving a weakly informative prior will not be the focus of this thesis. Instead, we will rely on the work of others.

However, we will mention that choosing a weakly informative prior does not have to be a complex endeavour and there are methods that help us with this. The most popular such method are *prior predictive checks*, these are similar to *posterior* predictive checks which we will describe in the next section. Simply put, prior predictive checks can help us to confirm that the prior(s) does what we expect it (them) to do. That the prior does what we think it does is usually trivially the case for simpler models but might not be obvious for more complex model structures where the interdependencies between priors could have unpredictable consequences [11]. So, to construct an appropriate weakly informative prior we would start with one of the two bullet points outlined above and add or remove information by trial and error until we get something that accurately represents our prior beliefs. First then will we look at data. This is however beyond the scope of this thesis but for the interested reader we refer to [11, chapter 4 & 7.3], [9] and [1, chapter 2].

A good prior can compensate for weak data [1, section 2.9]. This could be both a strength and weakness of Bayesian statistics and has historically been a point of contention from non-Bayesian's [14]. The choice of prior is important and could have a large impact on the posterior but as the data size increases the prior becomes less important. Asymptotically under some regulatory conditions (notably that the likelihood is a continuous function) the posterior will be dominated by the likelihood, no matter how strong or weak the prior is [1, section 4.2 & appendix B]. This result is illustrated in figure 1. Both priors in the figure are beta distributed. The weak prior is $\text{Beta}(2, 2)$ and the strong is $\text{Beta}(50, 50)$. "Data" here are binary trials that are generated deterministically so that we observe a 90% success rate over $n = 10$ or $n = 1000$ observations.

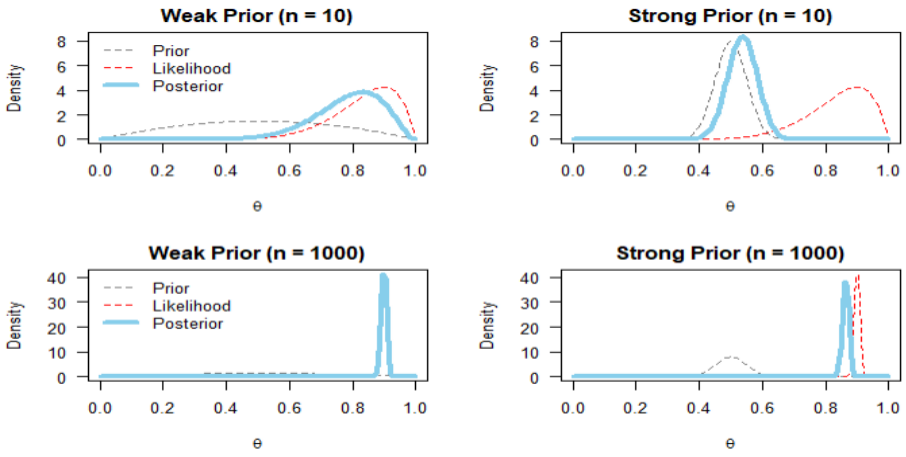

Figure 1: Weak and strong conjugate (beta distributed) priors for the same likelihood function of a binomial experiment. We vary the prior between columns and the number of observations (n) between rows. We assume a 50% success chance (for both priors) and always observe a 90% success rate, just with a different number of observations. The likelihood function in the lower left graph is completely covered by the posterior.

2.2.2 Bayesian Logistic Regression

Recall from section 2.1 that an ordinary logistic regression is given by

$$Y_i|\pi(\boldsymbol{x}_i) \overset{ind.}{\sim} \text{Bernoulli}(\pi(\boldsymbol{x}_i)) \quad i = 1, ..., n,$$

where Y_i is a binary outcome random variable and $\pi(\boldsymbol{x}_i)$ the logistic function in equation (2.1). Fitting the parameter values was done by finding the MLE. Bayesian logistic regression is the natural Bayesian extension of ordinary logistic regression. We assume a prior distribution for each parameter and use Bayes rule to get a posterior distribution

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood}$$

The likelihood function is the same as that described in section 2.1 but because of the prior we are no longer focusing on the MLE.

The prior distribution does not have to involve dependence between the parameters, instead we can assume prior independence with the understanding that the model can be reparameterized in places where prior correlation is appropriate [1, chapter 16]. Let $X = \{\boldsymbol{x}_i\}_{i=1}^n$ and $Y = \{Y_i\}_{i=1}^n$ refer to explanatory and response data respectively. With prior independence we end up with the following posterior:

$$\underbrace{f(\alpha, \boldsymbol{\beta}|Y, X)}_{\text{Posterior}} \propto \underbrace{\prod_{i=1}^{n} f(y_i|\boldsymbol{x}_i, \alpha, \boldsymbol{\beta})}_{\text{Likelihood}} \times \underbrace{f(\alpha) \prod_{j=1}^{p} f(\beta_j)}_{\text{Prior}},$$

where β_j, $j = 1, ..., p$ are the model parameters and $(\boldsymbol{x}_i, y_i)$, $i = 1, ..., n$ are the observations.

Some thought should be given when choosing a prior for this model, considering that small discrepancies on the logit scale can magnify into unreasonable estimates when transforming back to probabilities. Specifically, if the predictors are scaled reasonably it is hard to imagine a situation where any of the the logistic regression coefficients will be greater than 5 in absolute value (if the corresponding explanatory variable is increased by 1, this would move the probability from 0.50 to 0.99). By using a weakly informative prior we can include just enough information to regularize the extreme inferences that are possible with noninformative priors (or maximum likelihood methods more generally) [1, chapter 2]. Choosing a weakly informative prior will also prevent *separation*, which is a surprisingly common problem when applying logistic regression to real data [1, section 16.3].

Gelman et al. [1, section 16.3] recommends Students-t distribution as a baseline prior in logistic regression. The reason is that we can choose the degrees of freedom and scale in such a way as to constrain the coefficients to lie within a reasonable range while still providing minimal prior information and allowing for robust inference. If we decide to use a t-distribution we still need to choose its parameters. One way of doing this is to consider the baseline case of a single (Bernoulli) trial with probability $p = \text{logit}^{-1}(\theta)$. The corresponding likelihood is $e^{\theta}/(1 + e^{\theta})$, which looks like the cumulative distribution function of a t-distribution with scale hyperparameter 2.5 and 7 degrees of freedom [10]. For the intercept, α, a higher scale value might be justifiable. If we use the same hyperparameter values for the intercept we will be saying that we expect the success probability to be between 1% and 99% when all the inputs are averaged. This is often a reasonable assumption but there might be situations where it is not, like modelling rare events.

Separation refers to the case when one predictor (or a linear combinations of predictors) perfectly predicts some subset of discrete data. When this happen the MLE in ordinary logistic regression can be undefined or unreasonable, see [1] section 16.3 for an example.

2.2.3 Posterior predictive check

All theory in this section is from [1, section 6.3].

We emphasise that the posterior is an actual distribution. Posterior predic-

tive checks (PPCs) is a popular way of assessing if data generated from the model look similar to observed data. This is a self-consistency check, any observed discrepancy can be caused either by model misfit or pure chance.

Posterior predictive checking builds upon replicating data with the *posterior predictive distribution*

$$f(x^{\text{rep}}|x) = \int_{\Theta} f(x^{\text{rep}}|\theta) f(\theta|x) d\theta \tag{2.6}$$

and then comparing these replicates with observed data using *test quantities*. Here we used x^{rep} to specify that this is replicated data.

Test quantities are scalar summaries of some aspect of the data we wish to check. These can either be frequentist style test statistics, $T(x)$, which only depend on data or $T(x, \theta)$ that depend on both data and a parameter value, usually the maximum a posteriori probability or some prior belief. A Bayesian p-value is the probability of our replicated data being more extreme than observed data, or more formally

$$p = Pr(T(x^{\text{rep}}, \theta) \geq T(x, \theta)|x) \tag{2.7}$$

and equivalently for $T(x)$. A model is suspect if the observed value has a tail-area probability near 0 or 1, which indicates that it is unlikely to simulate the observed data from the posterior. To make comparison between p-values easier we usually convert lower tail p-values, i.e. p-values with probability more than 0.5, to upper tail p-values by changing the sign in equation (2.7). This means that a bad fit will have a p-value close to 0.

The probability in (2.7) is expressed in terms of the the joint distribution $p(\theta, x^{\text{rep}}|x)$. In more detail, it can be written as

$$p = \iint I\big(T(x^{\text{rep}}, \theta) \geq T(x, \theta)\big) p(x^{\text{rep}}|\theta) p(\theta|x) dx^{\text{rep}} d\theta,$$

where I is the *indicator function*. Although in practice this integral is seldomly evaluated, instead one usually uses simulation. This is very convenient because if we already have N simulations from the posterior, perhaps from a MCMC method such as the Metropolis-Hastings algorithm (more on this in section 2.3.1), then all we need to do is to draw one $x^{\text{rep},t}$ from the predictive distribution (2.6) for each simulated θ, here denoted θ^t, for $t = 1, \ldots, N$. The estimated p-value is then given by comparing the realized, observed test quantities $T(x^{\text{rep}}, \theta^t)$ with the predicted, simulated test quantities $T(x^{\text{rep},t}, \theta^t)$ and looking at the proportion of predicted test quantities from N simulations that meet or exceed the realized, observed values.

We get simulations from the posterior for free when training a model using an MCMC method. So, performing PPC is thus very cheap, computationally.

2.2.4 Hierarchical models

With Bayes rule, equation (2.4), we introduced the most basic Bayesian model. This model has two stages, one for the likelihood, $f(x|\theta)$, and one for the prior, $f(\theta)$. But, just as the likelihood depends on θ the prior might depend on another parameter ϕ. We would then get a three stage model with *hyperparameter* ϕ. Using conditional probabilities, the joint posterior distribution for a three stage model can then be written as

$$f(\phi, \theta|x) \propto f(x|\theta, \phi)f(\theta, \phi) = f(x|\theta)f(\theta|\phi)f(\phi), \tag{2.8}$$

with the last equation holding because the likelihood, $f(x|\theta, \phi)$, only depends on θ; whereas ϕ affects x only through θ. Of course, there is no reason ϕ cannot itself depend on another parameter η and so on. Specifying a model over several levels like this is referred to as *hierarchical modeling*, where each parameter is a level in the hierarchy [1, chapter 5]. Note that each parameter will have its own prior distribution.

We typically seek the marginal posterior distribution of the first stage parameter, which is derived by integrating the joint posterior distribution over all hyperparameters. For equation (2.8) that would be

$$f(\theta|x) = \int f(\theta, \phi|x)d\phi,$$

but for an m stage hierarchical model we would have to evaluate an $m-2$ dimensional integral [1].

In practice, hierarchical models are used when data is grouped in some way, but these groups can still be assumed to come from the same population distribution. Non-hierarchical models are generally not suited for such situations. With few parameters they tend to underfit large data sets and conversely overfit[1] with too many parameters. Because hierarchical models introduce randomness at each level and structure some dependencies into the data from the population distribution, it is possible to use enough parameters to fit the data without overfitting. In fact, Bayesian hierarchical models can often be fit with more parameters than there are data points without overfitting. It does this by implicitly controlling the amount of pooling at each level [1].

To understand the concept of pooling it is easiest to think in terms of a frequentist model. Assume we want to create a model that predicts a person's height based on some parameters and that one additional parameter ϕ is "country of origin". *No pooling* would fit everything together in one model. This might not be optimal if we think that there might be some significant differences between countries. Another option would be to fit one model for each

[1] Overfitting is when a model fits the existing data well, specifically the data used to train the model, but produces inferior predictions for new data.

country, this is the *complete pooling* approach. This could be an improvement but now we are "wasting" a lot of data that could potentially include valuable information. A third option would be *partial pooling* which is a mix of the previous two approaches. How that is achieved in a frequentist model is beyond the scope of this thesis but in a Bayesian model partial pooling follows by definition. This is because the model will automatically apply less pooling when the estimated population variability is high [1, section 5.4]. For example, in equation (2.8), if the variance of ϕ is low then $f(\phi)$ will always be constricted to a small area around its mean and the conditioning in $f(\theta|\phi)$ will not be that restrictive. This means that we might have defined a three stage model, but it still behaves like a two stage model. A similar thing happens in more complex model structures. For the model we will introduce in section 3.3 each θ (there called α) will depend on several sub-parameters, some of which might be small and have little impact while others have a larger impact (i.e. partial pooling).

2.3 Sampling from the posterior

When applying Bayesian inference to more complex models a potential problem that occurs is performing the integration necessary for the normalizing constant, $f(x)$, in Bayes theorem. In some problems, such as for logistic regression, exact Bayesian inference is intractable [4, section 4.5]. This is because evaluating the posterior distribution in a logistic regression model would require normalization over a product with a prior and a likelihood function that itself is a product of logistic sigmoid functions, one for each data point. In other situations, computing the exact posterior distribution might be possible but infeasible. Instead, we usually use approximate methods.

There are many methods one can use to approximate the posterior distribution. Each one of them has its own advantages and disadvantages. Broadly speaking, they all fall into two categories; either finding approximations of the integrals or avoiding the integration altogether by simulating directly from the posterior distribution (Monte Carlo methods) [3, chapter 8]. Chapter 10-13 in Gelman et al [1] describe some of the most popular methods.

For a Bayesian Logistic Regression (BLR) model specifically, several different methods are identified in the literature, such as Laplace approximation [4] or expectation-maxmimization [1, section 16.3]. But, the most popular approach is some kind of Markov chain Monte Carlo (MCMC) method.

MCMC methods have two main advantages over other methods, these are: 1) Unlike some numerical approximation methods, MCMC methods simulate the full posterior distribution and thus enable full Bayesian inference and 2) Unlike simple Monte Carlo methods, MCMC methods work well for distributions in high-dimensional spaces, thus being able to overcome the *curse of dimensionality* even when the regions of high probability are unknown to us beforehand [8, chapter 1 & 2]. Informally, the curse of dimensionality states that the volume

of a high dimensional space becomes so vast that observations in that space is sparse in comparison, which leads to problems of statistical significance. To overcome this, data size must grow exponentially with the number of dimensions. MCMC methods combine a random search (the Monte Carlo aspect) with intelligently moving around the posterior distribution, but in a way that does not depend on the starting point but rather on the current location (the Markov chain aspect).

In this section we will introduce one of the most popular MCMC methods, the Metropolis-Hastings (MH) algorithm. We will then discuss how to check for convergence, which is necessary to do after a simulation run because we are not guaranteed to converge after a finite number of iterations [1]. Finally we will implement and test the MH algorithm on simulated data.

2.3.1 Metropolis-Hastings algorithm

We will now introduce the MH algorithm, following (but simplifying) the outline laid out in [1, section 11.2].

1. Set $t = 0$ and choose a starting point θ^0. This starting point can be arbitrarily chosen as long as there is a positive posterior probability at that point.

2. Generate a new sample, θ_{new}, by using an asymmetric proposal distribution, $q(\theta_{\text{new}}|\theta^t) \neq q(\theta^t|\theta_{\text{new}})$.

3. Calculate the following quantity, know as the ratio of densities:

$$r = \min\left\{\frac{f(\theta_{\text{new}}|x)}{f(\theta^t|x)} \cdot \frac{q(\theta^t|\theta_{\text{new}})}{q(\theta_{\text{new}}|\theta^t)}, 1\right\}$$

4. Draw a sample u from the uniform distribution, $U(0, 1)$.

5. If $u < r$ set $\theta^{t+1} = \theta_{\text{new}}$, otherwise set $\theta^{t+1} = \theta^t$.

6. Set $t = t + 1$. If $t < N$ (the number of desired iterations of the algorithm) return to step 2, otherwise stop.

We have to make a choice of proposal distribution before we start. The ideal proposal distribution would be the posterior distribution, then the ratio r would always be 1 and each iteration would be an independent draw from the posterior [1]. However, if we already have access to the posterior then we would have no need for this algorithm. So, a sub-optimal choice must be made. This choice is important as it determines the rate of convergence. Some general guidelines for choosing a good proposal distribution are outlined by Gelman et al [1, section 11.2], stating: 1) It should be easy to sample from the proposal distribution for any θ, 2) the ratio r should be easy to compute, 3) each new draw should generate a move over a reasonable distance in the parameter space and 4) we

should not reject draws too frequently. Points 3 and 4 are included to assure that our simulation trajectory converges towards the posterior distribution at a reasonable pace.

Each accepted iteration of the MH algorithm is one sample from the posterior. Together these samples define a chain of random variates drawn from a distribution that will converge to the desired posterior distribution, $f(\theta|x)$. And so, after a number of iterations, all following samples will be from the posterior. These first iterations, before convergence to the posterior, is called the *burn-in* period and it is common practice to discard them from the chain(s) as they could influence model inference, even after convergence has been achieved. How many iterations to remove from the output of the Markov chain depends on the situation, Gelman et al. recommends 50% as a conservative standard.

To extend the algorithm to higher dimension d we choose an initial starting value for each $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)$ component and use an d-dimensional proposal distribution $q(\cdot|\cdot)$ [1].

For an hierarchical model the simulation is performed in two steps. For the three stage model in (2.8), we first note that the parameter of interest, θ, can be simulated by conditioning on the data and the hyperparameter, ϕ. Conditioning on parameter values like this is a special case of the MH-algorithm known as a *Gibbs sampler*. However, the hyperparameters are not known to us beforehand and must be simulated using MH. Therefore, we first need to simulate ϕ, and then condition on ϕ and data when simulating θ [1, section 5.3, 11.6 & 12.3]. For a discussion about the Gibbs sampler see *Gelman et al.*, [1], section 11.1.

2.3.2 Checking for convergence

MCMC methods are only guaranteed to accurately describe the posterior distribution asymptotically. Therefore, when we stop the algorithm we need to perform (non-)convergence diagnostics to determine if the chain has reached stationarity. Strictly speaking we cannot prove that the chain has reached stationarity, but we have a few tools that can tell us if it **has not**. If they all indicate convergence we can be fairly certain that the chain has converged [8, Chapter 6]. The reason we cannot prove convergence is that all assessment is relative to the domain explored by the chain. It is possible that a chain will struggle to escape a particular region of the posterior and thus give false positives. This is usually only a problem for more complex models and it can be mitigated by simulating several independent chains, each with random starting values from the parameter space. At convergence, we would expect these chains to have roughly the same variance and have mixed well. With "mix" we mean that one chain has not explored a region that the other(s) have not [1, section 11.4].

There are many methods available to assess convergence, some of them are specific to a special kind of MCMC algorithm. We will introduce the three most popular ones that can be used for any MCMC method, these are: The Gelman-Rubin test, effective sample size and trace plots.

The Gelman-Rubin test works by simulating several chains and assessing if the within chain variance is the same as the variance between chains. We start by simulating a number of chains and then split each of them in half (after removing the burn-in period). If we have reached convergence then both halves of one chain, and the halves from different chains, should have mixed well [1, section 11.4]. After splitting we will have m chains in total, each with n iterations. We label these simulations as ψ_{ij} $(i = 1, ..., n;\ j = 1, ..., m)$. We can now compute the within (W) and between (B) chain variances as follows:

$$B = \frac{n}{m-1}\sum_{j=1}^{m}\left(\bar{\psi}_{\cdot j} - \bar{\psi}_{\cdot\cdot}\right)^2 \qquad \text{and} \qquad W = \frac{1}{m}\sum_{j=1}^{m} s_j^2$$

where

$$\bar{\psi}_{\cdot j} = \frac{1}{n}\sum_{i=1}^{n}\psi_{ij}, \quad \bar{\psi}_{\cdot\cdot} = \frac{1}{m}\sum_{j=1}^{m}\bar{\psi}_{\cdot j} \quad \text{and} \quad s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\psi_{ij} - \bar{\psi}_{\cdot j}\right)^2.$$

Let y refer to data. To get an estimate of the marginal posterior variance $\widehat{\text{Var}}(\psi|y)$ we use a weighted average of W and B, namely

$$\widehat{\text{Var}}(\psi|y) = \frac{n-1}{n}W + \frac{1}{n}B.$$

This estimate is unbiased under stationary conditions (if the starting distribution equals the target distribution) or in the limiting case when $n \to \infty$ [1].

We can now monitor convergence by estimating the scale at which the current distribution ψ could be reduced if we continued simulating. This scale reduction is estimated by

$$\hat{R} = \sqrt{\frac{\widehat{\text{Var}}(\psi|y)}{W}}, \tag{2.8}$$

which will decline to 1 as $n \to \infty$. So, if $\hat{R} \approx 1$ we have good reasons to believe that our parallel chains have converged [1].

Because we are working with Markov chains, the samples from each one of them will typically be autocorrelated [1, section 11.5]. This means that simulation draws are correlated. Compared to independent draws, correlated simulations will increase the uncertainty when estimating quantities of interest from the target distribution. The *effective sample size* (ESS) is an estimate of how many independent samples it would take to achieve the same level of precision. Ideally, the ratio between ESS and the total number of simulations N should be

high. If it is low then we might be suspicious of the result. Unfortunately, there is no general rule for what constitutes a low/high ratio. Gelman el al. suggests running the simulation until we have an ESS of at least $5 \cdot m$ for each predictor, and run it for longer if we require more precision [1]. However, it is worth noting that for models with few predictors (less then 10) we can generally expect far higher ESS.

How ESS is estimated is somewhat involved. For a rough estimate: note that B/mn estimates the variance of $\bar{\psi}_{\cdot\cdot}$. Suppose now that we are able to take n_{eff} independent samples from the posterior. The variance of the mean of this sample would be $\text{Var}(\psi|y)/n_{\text{eff}}$, where we estimate $\text{Var}(\psi|y)$ as $\widehat{\text{Var}}(\psi|y)$. Equating these two estimates gives

$$n_{\text{eff}} = \frac{mn \cdot \widehat{\text{Var}}(\psi|y)}{B}.$$

This equation is only valid asymptotically or under the assumption that the whole target distribution has been explored (but not necessarily exhaustive enough for convergence) [1, section 11.5]. With finite Markov chains we can get more accurate variance estimates by explicitly estimating the autocorrelations between draws at different time lags, see [1, section 11.5].

2.3.3 Simulation study

We will now implement a Metropolis-Hastings algorithm and use it to illustrate some of the concepts introduced in the previous sections. We will use an example that is partly inspired from sections 2.1 and 6.3 in *Gelman et al.* [1].

Consider a sequence of coin tosses (or binary trails in general) $\boldsymbol{y} = (y_1, y_2, ..., y_n)$, which can be modelled as independent and identically distributed (i.i.d.) Bernoulli trails. Now let x denote the total number of heads in our data and let θ denote the probability of heads for each trial. The likelihood function is then represented by the binomial distribution, i.e

$$p(x|\theta) = \text{Bin}(x|n, \theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x}.$$

Using a weakly informative prior distribution where we assume θ is uniform on $[0, 1]$ gives us the following posterior

$$f(\theta|x) \propto f(\theta) p(x|\theta) \propto \theta^x (1-\theta)^{n-x}, \tag{2.9}$$

which we identify as the (unnormalized) $\text{Beta}(x+1, n-x+1)$ distribution.

This is the general form of the posterior, however in practice we would work with observed data. Here we will just set $n = 20$ and $x = 7$, i.e. 20 coin flips with 7 heads. We will now implement the MH algorithm with a random walk proposal distribution where $\theta_i = \theta_{i-1} + N(0, 0.4)$. We will choose a random value in $[0, 1]$

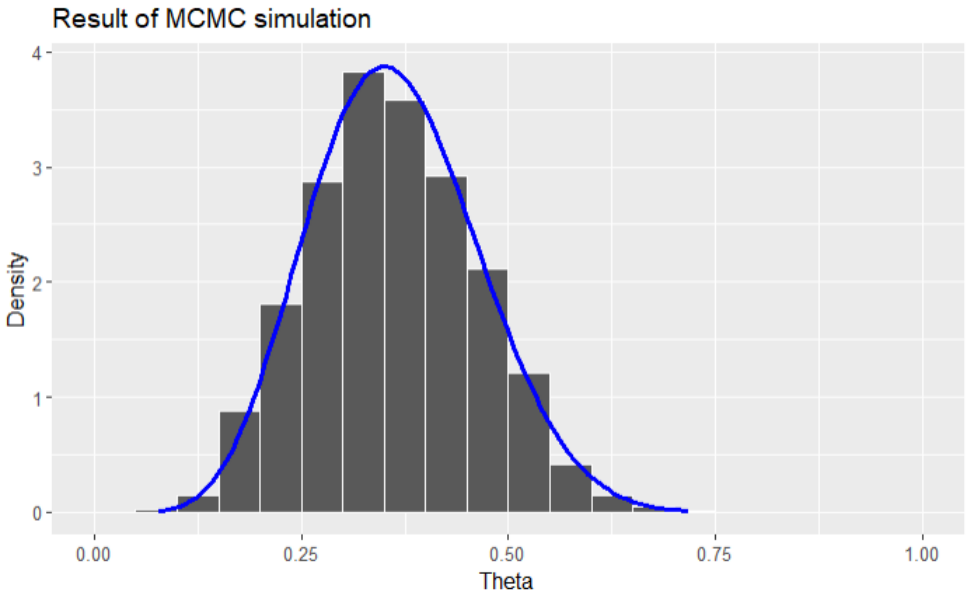

Figure 2: 2500 draws (after removing burn-in iterations) from the Beta(8,14) posterior in (2.9), with $n = 20$ and $x = 7$, using a Metropolis-Hastings Markov chain. The blue line is the Beta(8, 14) distribution.

as our starting point, then make 5000 iterations and finally discard the first 2500 of these iterations as the burn-in period. The result is displayed in figure 2, the blue line is the Beta(8, 14) distribution.

The result of figure 2 is the one we expected. When working with a distribution this "simple" we can expect fast convergence, even with a sub optimal proposal distribution. However, for completeness we will perform a convergence check. To do this we start by simulating another chain in the same way as we simulated the first one. We now have 2 independent chains, each with 2500 samples (after removing the burn-in periods). By splitting the chains in two we now have a total of $m = 4$ chains with $n = 1250$ samples each, and computing $\hat{R}$ from (2.8) gives $\hat{R} = 1.00191$. And the effective sample size ratio for both chains is over 80% so both methods indicate that our Markov chains have converged to the posterior.

Furthermore, let us assume that tossing the coin 20 times gave the following outcome:

$$1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,$$

where 1 is heads and 0 is tails, so the number of heads is still 7. Performing a posterior predictive check on $T(x, \theta = 0.5)$, where $T(\cdot, \theta = 0.5)$ is the number of heads on $m = 10000$ runs ($n = 20$ draws each run), where we assume $\theta = 0.5$, gives us a p-value of about 20.7%, as illustrated in figure 3(a). From this test

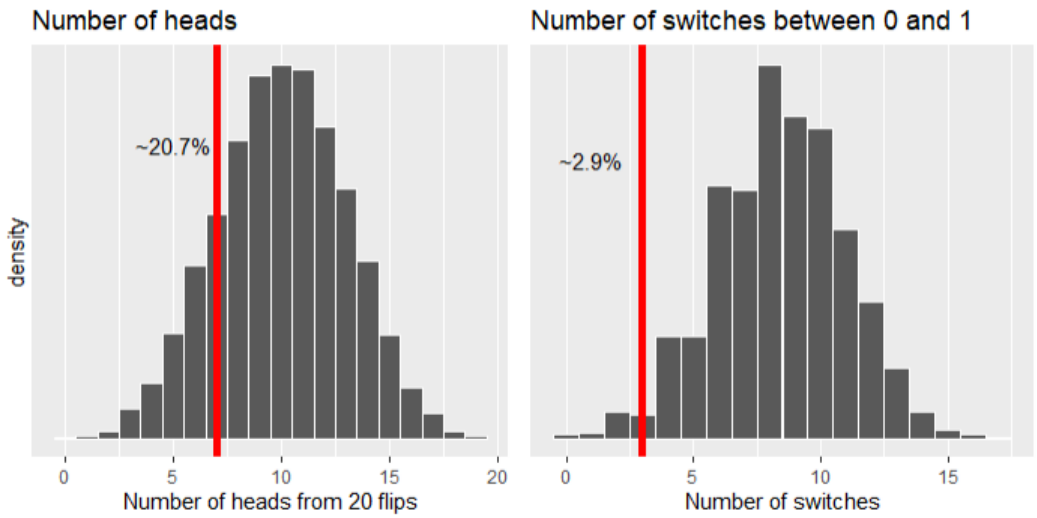

Figure 3: (a)(left) A display of the density of the number of heads and (b)(right) a display of the density for the number of switches between 0 and 1. Both plots contain 10000 samples from the same posterior. The red cutoff line represent the observed values with the corresponding lower tail probabilities obtained from the predictive distributions.

quantity alone we do not have any reason to be sceptical about our model. However, we notice that our observations look highly autocorrelated

(same type of outcome stack up). Since we assumed a model with i.i.d Bernoulli trails this is not a result we would expect. If instead we let $T(\cdot)$ be number of switches between 0 and 1 we get a p-value of about 2.9%, which is much less likely. This result is displayed in figure 3(b) and it is pretty strong evidence against the i.i.d assumption of our model, although it can still be caused by random chance. Notice that the first test quantity depends on both data and a hypothesized parameter value while the second only depends on the data. The distribution in the first test quantity is thus $\text{Beta}(11, 11)$ and in the second it is $\text{Beta}(8, 14)$ (the posterior). We could think of the first test as a goodness of fit test, where we want to examine if the coin is fair, while the second test examines some model assumption(s).

2.4 The Elo rating system

The purpose of any rating system is to provide a framework to compare different players/teams based on whatever is being rated. A good ranking system should do two things

1. Evaluate performance of a player/team on some sort of scale so that competitors might be listed in the probable order of their skills.

2. Provide an estimate of two competitors' relative skills.

The *Elo rating system*, originally developed to rank chess players by Arpad Elo [7], is a widely used ranking system. The system works by first assigning each player a starting skill estimate (or elo), usually 1500 and then adjusting the player's rating after each game depending on the strength of his opponent and the outcome of the game [7].

The formula used for elo adjustment is given by

$$R_A^{(t+1)} = R_A^{(t)} + K(I_A - p_A), \tag{2.10}$$

where $R_A^{(t)}$ is the rating of player A before game number t, K is a constant which determines how much elo should be updated after each match (usually set to 32), I_A is the indicator function of A winning the game and p_A is the elo-estimated probability of player A winning, calculated by

$$p_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}. \tag{2.11}$$

This probability function is defined in such a way that a 100 elo difference $R_A - R_B$ translates into a 64% win probability, a 200 elo difference gives approximately a 76% win probability and a 300 elo difference an 85% win probability [7].

From (2.10) we see that if player A wins the match he is awarded less elo if he is the favorite and more if he is the underdog, and reversely if he loses. It is interesting to note that $R_A^{(t)} + R_B^{(t)} = R_A^{(t+1)} + R_B^{(t+1)}$, i.e that the losing player loses as much elo as the winner wins.

3 BLR on Tennis data

Bayesian methods are becoming more and more popular in sports analytics. According to Santos-Fernandez et al. [6] a similar number of publications appeared between 2013 and 2018 as the number published in the previous three decades before 2013. It is easy to see why. Better computer processing power and advances in MCMC methods have greatly reduced many of the disadvantages of Bayesian data analysis.

A bit simplified, we can summarize Bayesian data analysis according to the following three steps:

1. Specify a posterior distribution.
2. Draw from this posterior using MCMC.
3. Evaluate how well the model fits the data.

A possible fourth step would be to revise the model if we are unsatisfied with the results in step 3. We would then ideally use information from the first model to refine it and find an improved second model.

In this section we will use data from official tennis matches to create a logistic regression model that tries to predict the winning player. We will then evaluate how well our model fits the data (with posterior predictive checks), attempt to improve the model and finally look at how well our model performs on test data.

3.1 Preliminaries

3.1.1 Data

The data is taken from "Tennis-Data" [5] and contains all (men's) official tennis matches from 2015 up to December 2021, in total about 15,000 data points. There are a few categories in the data, including the following: Players, winner, tournament (name and location), match date, series (Grand Slam, Masters, ATP500 or ATP250), tournament round, court surface, best off and betting odds from a few different bookies.

Court surface refers to the surface material, which will affect how the ball ricochets; some players might be better on one surface than another. We have not included a way to measure this relative skill in our data, so we will ignore it. A series can be seen as the tournament level. The highest level, where the best players in the world compete, is "Grand Slam", followed by "Masters", "ATP500" and finally "ATP250". Every tournament belongs to one series. Best of refers to how many sets each game consists of, either 3 or 5. The winner of the game is the player that wins a majority of these sets. A game cannot end in a draw.

We will only use odds from one specific bookie, Pinnacle. This is because, out of all the bookies in our data, it is the only one that uses price discovery and does not restrict winning gamblers from placing bets. Henceforth when we say "bookie odds", we mean Pinnacle odds. All odds are closing odds, meaning that they are the last odds offered by the bookie before the matches begin. These closing odds are supposed to best represent the real win probabilities for these matches (when converting them to probabilities). Henceforth we will call this variable "bookie probabilities". We prefer working with probabilities over odds because probabilities are restricted in $[0, 1]$ and should thus give more stable results when using numerical methods.

3.1.2 Data preparation

Elo rating was not part of our original data set but can easily be calculated by repeatedly applying equation (2.8) after each game. We set each player's

starting elo at 1500 and use the standard $K = 32$ value. Elo is updated after each game. So, each player's new elo can only be seen in the next observation for this player. The data is mostly sorted by date, with exceptions only for overlapping tournaments. Since one player cannot play in two tournaments at the same time we do not need to sort the data.

When the elo rating has been calculated for each match we split the data into three parts: 1) 2015 - 2016, 2) 2017 - 2020 and 3) 2021. The first part is the time it takes to get a reasonable elo estimate for each player. This part of the data will be discarded when training our model. Every player started in 2015 with the same elo, so including this data for training could negatively effect our final result. Only the second part of data is used to train the model and the final third part will be used as test data.

For each individual match, we choose one of the two players randomly. Our model will try to predict this player's win probability, but since a tennis match cannot end in a draw the complementary probability will be his opponent's chance to win. This is done to minimize the risk of introducing bias in the model. This is especially true if we use information from the future, for example if we consistently try to predict the winner's/loser's win chance. When we have a "focus player" we will calculate the elo difference between the two players. If the difference is positive then the player we choose is the favorite, otherwise his opponent is the favorite.

Bookies make money from the spread between their offered odds. This means that bookie probabilities do not sum up to 1, and the excess above 1 is the spread. This spread is not constant between different matches. How big the spread is can vary between matches, depending on the bookies' turnovers for these matches. The more popular Grand slam matches typically have lower spread. Therefore we choose to remove spread from the data as to not introduce bias and keep the probabilities of each game complementary to each other. This is done by calculating the spread for the match and removing half of it from each player's win probability.

The elo difference and the bookie probabilities are then standardized using the Z-score. That is, they are standardized by

$$Z = \frac{x - \hat{\mu}}{\hat{\sigma}},$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the estimated sample mean and standard deviation respectively. By doing this they will both have mean 0 and standard deviation 1, so we will more easily be able to compare their relative influence in the final model. Standardizing also has the added benefit of reducing simulation time.

	Mean	Standard error	Rhat	ESS ratio
Intercept (α)	-0.026	0.018	1.002	0.582
Elo difference (β_1)	0.276	0.025	1	0.964
Bookie probability (β_2)	0.877	0.059	1.004	0.481

Table 1: Summary statistics of a Bayesian logistic regression model with effect parameters "elo difference" and "bookie probability".

3.2 Fitting a BLR model

We will start with a simple model with the understanding that complexity can be added after model evaluation if (and where) it is needed. This model will only include two predictors: difference in elo rating and bookie probabilities. We expect elo to be a reasonably good predictor and bookie probabilities to be a very good predictor. We will assume prior independence between the effect parameters of these two predictors and choose weakly informative t-distributed priors with scale 2.5 and 7 degrees of freedom (baseline case). We expect the intercept to be located around 0 so we will choose the same prior for it as well. Our final model, following the notation of section 2.2.2, is:

$$\begin{aligned} Model: \quad & f(\alpha, \beta_1, \beta_2 | \boldsymbol{y}, \boldsymbol{X}) \propto f(\alpha) f(\beta_1) f(\beta_2) \cdot \prod_{i=1}^{n} f(y_i | \boldsymbol{x}_i, \alpha, \beta) \\ \text{Priors}: \quad & f(\alpha) \sim t_7(2.5), \\ & f(\beta_1) \sim t_7(2.5), \\ & f(\beta_2) \sim t_7(2.5), \end{aligned}$$

where $t_v(\sigma)$ is the t-distribution with scale σ and v degrees of freedom.

We now fit this model to the data using 2 Markov chains with 2500 iterations each. After removing the first 50% of iterations as the burn-in period we have 2500 total iterations left. Table 1 shows the results. Both $\hat{R}$ (Rhat) and the estimated sample size (ESS) indicate that the chains have converged. Since we do not have any reason to think otherwise, we will accept this result and move forward. Noting only that the coefficients, α, β_1, β_2, are all on the logit scale (and estimated with standardized data), we can see that bookie probability is the by far strongest predictor.

We will now perform various posterior predictive checks (PPC) to evaluate how well our model fits (training) data. This will be done by choosing data points of interest from the training set and using them to draw 2500 samples from the posterior for each data point with the same parameter values (as for these data points). We will then look at how likely it is to draw observed data from our model by looking at the proportion of posterior simulations that meet or exceed observed data. Recall that the p-value can be interpreted as the cumulative probability, according to our posterior, for a test statistic to meet or exceed

observed data. The test statistic we use here is just the mean.

By exploring the data it is reveled that a majority (about 55%) of observations are contained within a fairly small parameter interval with bookie probabilities 0.5 ± 0.2 and elo difference 0 ± 220 (the largest and smallest observed elo differences are 677 and -678 respectively). Only about 11% of the observations have a bookie probability lower than 15% or higher than 85%, even fewer observations if we also include tail elo differences. Because of this we are worried that when training the model on this data there will not be enough variance in the model to fit tail events, with bookie probabilities close to 0 or 1. This is known as *overdispersion* and arises because the variance in the Bernoulli distribution depends on the mean [2].

We will start our PPC by examining overdispersion. Figure 4a shows a PPC on the mean posterior probabilities $P(x^{rep}|x)$ in (2.6), of observations x^{rep} for which bookie probabilities are less than 15%. The corresponding lower tail p-value is 0.002, and we can also clearly see from the figure that our model overestimates the win probabilities for these events. The histogram here, and in future PPC plots, is constructed by looking at the simulated means for each data point. Doing the same PPC test on bookie probabilities less than 20% returns a p-value of 0.2114. This indicates that we indeed have overdispersion and that the posterior distribution's bell curve is quite steep. Interestingly, if we do the same test when the elo difference is less than -250 we get an upper tail p-value of about 0.146, suggesting that our model underestimate these events[2]. Figure 4b shows the aggregate case, when we look at both bookie probabilities and elo difference. This result is slightly better, but still bad. What happens here is that the underestimated win chances of elo differences shift the posterior to the left but, since bookie probablity is a stronger predictor, not by much. The posterior distribution is approximately symmetric so looking at upper tail observations (i.e, bookie probability $> 85\%$ and/or elo difference > 250) gives analogous results. For now, we will just take note of this but we will discuss it further in later sections.

We have overdispersion in our model but it fits the data well in regions where the bulk of data is. For example when bookie probability is between 25% and 75% the p-value is 0.4176. If we include elo difference being between -200 and 200 then the p-value is 0.3116. Furthermore, widening one of these intervals and tightening the other still gives good p-values. Over the whole data set the posterior mean is consistently within a range of $\pm 0.25\%$ from the observed win proportion, even when the "focus" player is randomly reshuffled but the same model is kept. Figure 4c shows posterior draws for the whole data set.

There are a few predictors left in our data that we did not use in the model,

[2]Although this could be caused by random chance. That is, even if our model is correct, we would expect to see such discrepancies occasionally just by chance

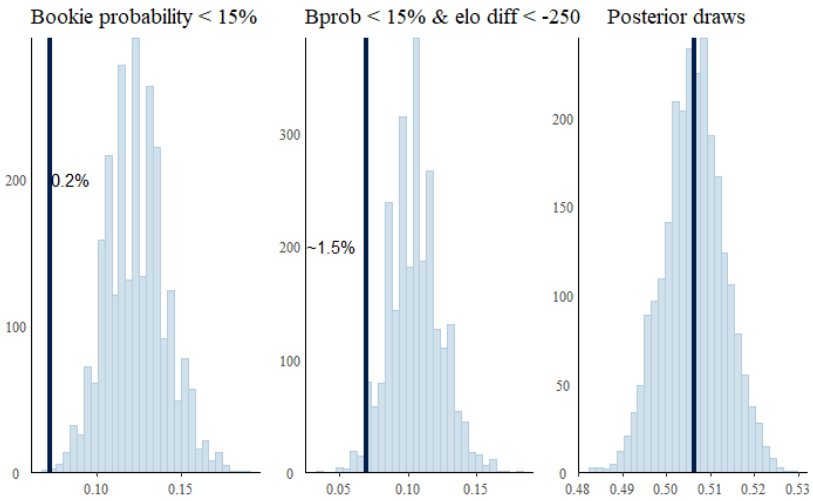

Figure 4: (a)(left) PPC of the mean (the win probability) for games such that the bookie probability is less then 15%. (b)(middle) PPC of the mean when the bookie probability is less then 15% and the elo difference is less than -250. (c)(right) Posterior draws of win probabilities, without any restrictions on the predictors. The vertical solid bars refer to the fractions of wins, for each data set, and the p-values are PPC probabilities for the regions to left of these bars.

notably "best of" and "series". Instead, we made the assumption that these are unimportant predictors and if included they would not effect fit of the final model. If our assumption is true then PPC on this expanded data should be similar to the PPC for the data set of figure 4c. We will now examine this assumption.

We reason that, all else being equal, best of 5 sets games will favor the favorite because the variance will even out over longer games. Figure 5a shows a graphical PPC examining this where "favorite" is defined as having a bookie win probability over 70% and figure 5b shows results for the underdog ($< 30\%$ bookie win probability). The PPC for favorites is not great, but good enough. For underdogs however, it is terrible. Looking at the figures more closely we see that the average win proportion of favorites is around 83.2% while it is 12.0% for underdogs. We would expect there to be symmetry here, as it seems to be for the posterior draws (roughly speaking). Since we choose which player to focus on randomly the observed difference here is likely caused by chance. If we look at favorites and underdogs for the whole data set then the win proportion is about 80% and 20% respectively and the p-value is about 0.4 for both. So, we have symmetry over the whole sample but even if we are optimistic and assume that the p-value for underdogs should be as for favorites they still seem to differ from the data as a whole.

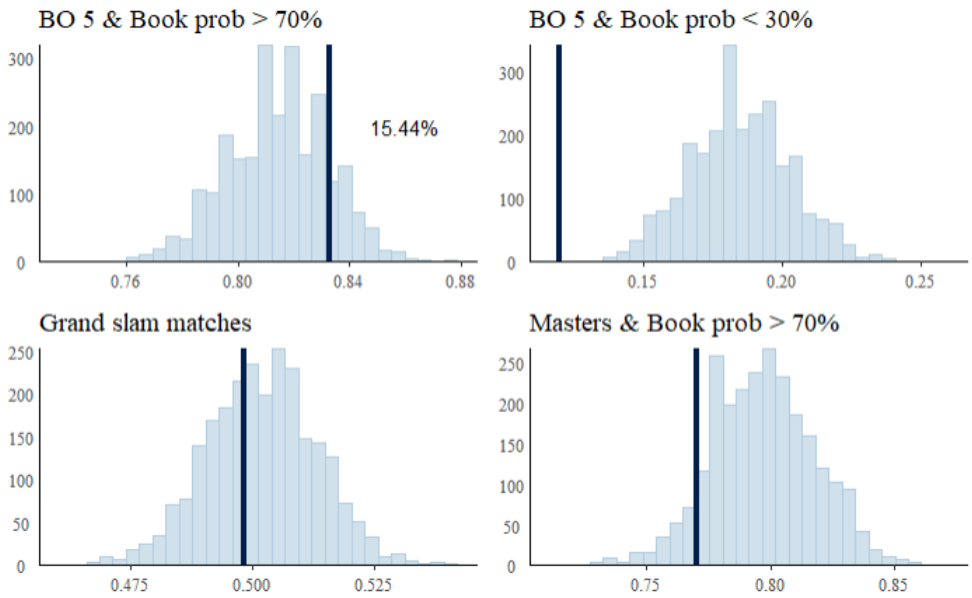

Figure 5: (Clockwise from top left)(a) PPC of the mean response (the win probability) of the favorite (bookie probability over 70%) for best of 5 sets matches. (b) The corresponding PPC plot for the underdog (bookie probability less than 30%). (c) PPC of the mean response (the win probability) for Grand slam matches. (d) PPC plot for favorites of Master (series) matches. The solid vertical bar of each subplot refers to the observed fraction of wins for the corresponding data set.

When looking at the data more closely it turns out that all Grand Slam matches are played best of 5, while the matches in other series (even finals) are played best of 3. This means that all information in the "best off" variable can also be captured in "series". Figure 5c shows a graphical PPC on the mean of all Grand Slam matches (i.e, all best of 5 matches). The posterior draws here are shifted slightly to the right and in other series it is shifted to the left, and the combined posterior draws correspond to the result in figure 4c. This makes us think that our model will overestimate the favorite's win chance in other series to compensate for the underestimation in figure 5a. This is indeed the case, most notably for the "Masters" series, this is plotted in figure 5d. Because of this we have reason to believe that data is grouped by series.

3.3 Second model

We have identified two discrepancies in replicated data compared to observed data in the first model, these are: 1) Overdispersion and 2) Systematic differ-

ences between the different series. We will attempt to improve on the second point by introducing a hierarchical model. We will skip introducing a three stage model with series as sole grouping variable and instead use a model with four stages, where each "series" also have "tournament" sub-groups. We are not going to directly address the overdispersion problem but hopefully, the extra flexibility of this model could mitigate the effect of overdispersion. That is, if the within tournament variance is higher than between tournament variance then the model could be flexible enough to reduce overdispersion. However, this seems unlikely, given how professional tennis points are awarded[3].

We assume that the effect parameters of the predictors (elo difference and bookie probability) are the same for each group and that only the intercept changes. We will also assume that these effect parameters are independent and define the following model

$$
\begin{aligned}
Model: \quad Y_{ijk}|\alpha_{jk},\beta_1,\beta_2 &\sim \text{Bern}(\pi_{ijk}) \\
\text{logit}(\pi_{ijk}) &= \alpha_{ij} + \beta_1 X^{(1)}_{ijk} + \beta_2 X^{(2)}_{ijk} \\
\text{Series variability}: \quad \alpha_i|\alpha,\sigma &\sim \mathcal{N}(\alpha,\sigma^2) \\
\text{Tournament variability}: \quad \alpha_{ij}|\alpha_i,\sigma_i &\sim \mathcal{N}(\alpha_i,\sigma_i^2) \\
\text{Global priors}: \quad \alpha &\sim t_7(2.5) \\
\beta_1 &\sim t_7(2.5) \\
\beta_2 &\sim t_7(2.5) \\
\sigma &\sim \text{Exp}(1) \\
\sigma_i &\sim \text{Exp}(1)
\end{aligned}
$$

Here $X^{(1)}_{ijk}$ and $X^{(2)}_{ijk}$ are the explanatory variables that β_1 and β_2 are fitted on, $i = 1,2,3,4$ denote series, $j = 1,\ldots,n_i$ is the tournament (different number of tournaments depending on what series they belong to) and $k = 1,\ldots,n_{ij}$ are the matches within each tournament. α_{ij} is the intercept parameter which now is different for each series and tournament. The notation here is a bit messy, when we write $\alpha_i|\alpha,\sigma$ we mean that the series intercept α_i is conditioned on the global intercept and standard deviation parameters α and σ for each i. The tournaments, j, are those that belong to this series and the games, k, those that belongs to that tournament. The variance components σ and σ_i refer to the magnitude of random exponential noise, which adds randomness to each series, and all tournaments within series i, respectively. Exponential noise is used as prior for the variance components across levels because they weakly regularize towards zero and only express a rough notion of an average standard deviation [11, section 13.1].

[3]The system is ridiculously complex but very simplified: if one tournament is significantly easier then others in the same series then players would receive carrier and pay benefits by exploiting that, which in theory should diminish the gap.

	Mean	Standard error	Rhat	ESS ratio
Elo difference (β_1)	0.274	0.028	1.000	1.534
Bookie probability (β_2)	0.880	0.032	1.002	1.395
ATP250 (α_1)	0.008	0.040	1.001	0.472
ATP500 (α_2)	-0.014	0.048	1.001	0.521
Masters (α_3)	-0.037	0.053	1.001	0.482
Grand slam (α_4)	0.054	0.060	1.001	0.490
Shenzhen Open (ATP250)	0.001	0.052	1.001	1.208
BNP Paribas Open (Masters)	-0.029	0.052	1.002	0.766

Table 2: Summary posterior statistics of a Bayesian logistic regression model with effect parameters β_1 and β_2 for the two predictors "elo difference" and "bookie probability". Displayed are also intercept estimates $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ of the four different series. "Shenzhen Open" and "BNP Paribas Open" are two tournament intercepts, in parentheses are the series they belong to.

We fit this model the same way as before: 2 Markov chains with 2500 iterations each and 50% burn-in. In total there are 107 tournaments (across all series), which are too many to show in one table. Table 2 shows the summary for the effect parameters β_1 and β_2 as well as intercept means for all series and a couple tournaments. Most of the remaining tournaments intercept means are around 0 ± 0.02 but the biggest is 0.088 and the smallest -0.043. When looking at the different tournament intercepts by eye there does not seem to by be a significant difference between them, which indicates a small between group variance. If we compare the effect parameters in table 2 with the ones from model 1 in table 1 we see that they are very similar. This was expected because they were both fitted on the whole sample. Amongst all parameters, no $\hat{R}$ (Rhat) value is over 1.01 and the ESS ratio is never less than 0.35 with most values higher than that. From this we do not have any reason to doubt that the chains have converged. The ESS ratio are sometimes larger than 1, which means that there are some negative autocorrelation estimates in the corresponding samples [11].

Figure 6 shows some of the graphical posterior predictive checks, analogous to those that were shown in figure 4 and 5 for the first model. By comparing the plots in figure 6 with the same one in figure 4 & 5 we can see that they are very similar. The fit for different series seem to have improved slightly. Other then that, there are some small differences but these can be attributed to randomness. The same goes for other PPC plots not displayed here. This was also suggested to us from table 2. The effect parameters between models were similar (as expected) and many of the slope distributions had a small mean and standard error. So, there were no significant difference between the two models in terms of PPC.

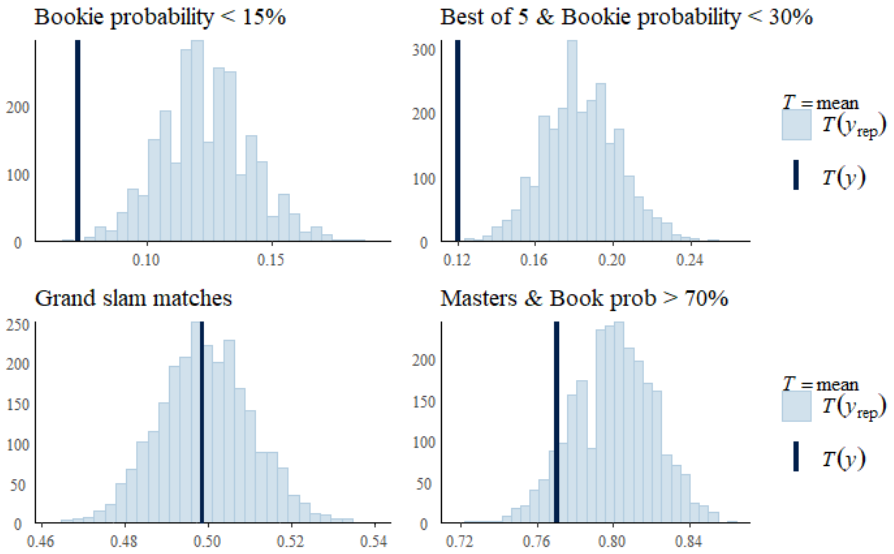

Figure 6: Analogous PPC plots as in figures 4 & 5 (for first complete pooling model) done instead with the hierarchical model (model 2).

3.4 Model evaluation & comparison

3.4.1 Evaluation metrics

In the previous section we were not completely satisfied with the PPC from the first model so we sought to improve it in the second hierarchical model (with lackluster results). However, as we previously mentioned: PPC is a self-consistency check, any observed discrepancy can be caused by model misfit or pure chance. In this section we will evaluate the models further and compare the results between models. We will use the following metrics: Prediction accuracy, Brier score, betting return and WAIC. All these will be performed on test data. This data was left out when training the model specifically for this purpose and it consist of 1631 matches played in 2021. For the rest of this thesis we will denote the first, non-hierarchical model, as "model 1" and the hierarchical model as "model 2".

Prediction accuracy (PA) is the proportion of correct classifications a model makes. We predict that player A will win if his estimated (model) win probability is over 50%. Another way to measure prediction accuracy is with *Brier score*, which is calculated by

$$\frac{1}{n}\sum_{i=1}^{n}(\text{forecast probability} - \text{outcome})^2$$

where n is the number of games in the sample. Like PA, the Brier score is bounded between zero and one. A high Brier score indicates bad predictions [12].

Betting return is the return (that is, the expected profit/loss) we would make if we used the model to place bets. This will be calculated by placing bets so that the potential return is 100 units. The return is calculated on odds with bookie spread. We will only place bets if the model estimates a 5% advantage over the bookie. The 5% number was chosen because the average spread between odds in the training data set was 2.56% and there is some uncertainty in our model predictions. We want the model to be reasonably sure about beating the bookies spread before placing a bet.

The *widely applicable information criterion* (WAIC) is given by

$$\text{WAIC} = -2\left[\log ppd - \sum_{i=1}^{n} \text{var}_\theta \left(\log p(x_i|\theta)|x\right)\right]$$

where $ppd = \prod_{i=1}^{n} p(x_i|x)$ is obtained from the posterior predictive distribution of equation (2.6), and the sum is the *penalty term*, known as p_{WAIC}. The penalty term reads as "compute the variance in log-probabilities for each observation i, and then sum up these variances to get the total penalty". We can think of p_{WAIC} as the "effective number of parameters" but one should be careful with this interpretation when working with hierarchical models. Because of how pooling works, increasing the number of parameters will not necessarily effect p_{WAIC} and it could even decrease it [11, section 7.4.2].

The justification for this metric is slightly beyond the scope of this thesis but the interested reader can read [11, section 7.4] for a (light) information theoretic justification, [11, section 7.2] for information theory in model evaluation more generally and [4, section 3.2] for a discussion about bias-variance trade off. For the purpose of this thesis it is enough to know that WAIC penalizes models with high variance and that those kinds of models tend to overfit data (because they have a large effective number of parameters and a small bias).

3.4.2 Analysis

Prediction accuracy, Brier score and average bet return for both models are displayed in table 3. For comparison, the table also displays the elo rating system and bookie odds. From table 3 we can see that both models outperformed the elo rating system but unperformed compared to the bookie on all metrics. In figure 7 we plot the cumulative return of the two models as well as the odds spread (blue line). The spread is the expected value of the betting return when placing random bets. Over the long run with random guessing we would expect prediction accuracy, Brier score and average bet return to be close to 0.5, 0.25 and -2.56% (average spread) respectively. From table 3 we can see that both

models outperform random guessing for all metrics, although no model is good enough to beat the spread.

Both models produce similar predictions. This was strongly indicated by the metrics in table 3, but especially apparent from the cumulative returns plotted in in figure 7, with the spread causing a negative return (that is, a negative expected profit) per game. Looking closely at the two curves of figure 7 we can see that both models often placed the same bets. Table 4 displays WAIC. Even here there is a small difference between WAIC values and their standard errors (SE) of model 1 and 2, which again implies nearly identical predictions. This means that the group intercept parameters in model 2 have been shrunk so much towards zero that they vary very little in the model. Furthermore, in model 1 all 3 parameters were "effective" (since $p_{\text{WAIC}} = 3$) while only 15.8 out of 109 parameters of model 2 were effective. Because WAIC is so similar p_{WAIC} here suggests that the different group intercepts in model 2 all had a small individual impact, rather than some being small and some large. We also indicated in table 2, where the largest group intercept was for "Grand slam" (series) matches, which in absolute terms still was small.

There are some insights to be gained in looking at where the two models differ from each other. Perhaps not as a performance diagnostic but for deeper model insights and as a consistency check. Both models had the same effect priors fit on the whole sample, but model 2 was allowed to have varying intercepts for each of the series/tournament groups. We know that both models made similar predictions. Looking at the predicted win probabilities more closely reveals that the average predicted probability difference (in percentage points) between the two models over the whole test sample was only 1.3%. But this figure increases to 1.5% when we just look at Grand slam matches. Moreover, when we look at the matches where bookie probability was between 0.2 and 0.8 (i.e., when the group intercept plays a bigger role) the average difference is 1.5% and almost 2% among these matches when we only consider Grand slam matches. This is consistent with our previous analysis and the results in table 1 and 2. So, even if the results from the "improvements" made in model 2 are somewhat disappointing the model does perform as we specified. There is simply not that much information in the groups, or if there is then it is dominated by information from the effect parameters, which perhaps is more likely.

Over the long run we would expect placing bets on a random player (i.e. random guessing) to have an average bet return close to -2.56% (the average spread). We saw from table 3 that both our models have a negative average bet return but they both beat random guessing. Some insights could be gained in examining why and how they beat the bookie's odds. This is not a trivial matter. Closing odds are derived by price discovery so to beat random guessing we need to derive odds that systematically exploit some market inefficiency. It is thus worth considering where and how our models diverge from the bookie's odds and examine the wider market implications, if possible.

	Model 1	Model 2	Elo rating	Bookie
Prediction accuracy	0.6517	0.6529	0.6143	0.6843
Brier score	0.2148	0.2151	0.2286	0.2026
Average bet return	-2.1879	-2.0081	-3.8535	-

Table 3: Summary of different models' performance measures.

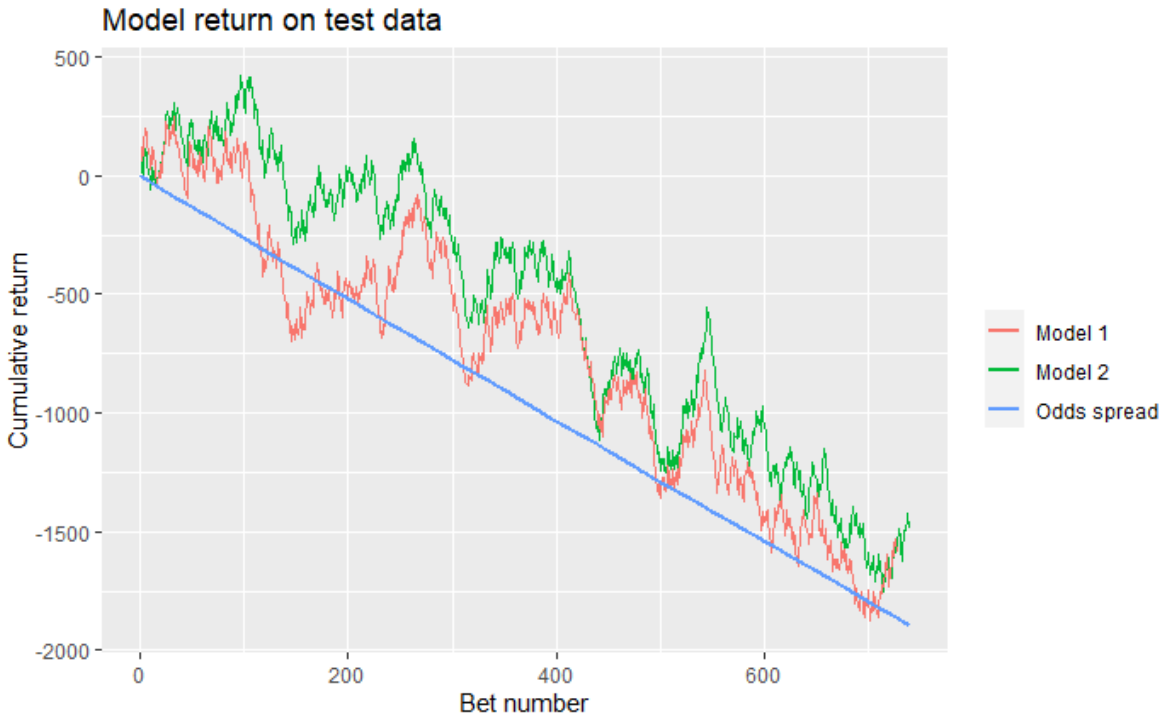

Figure 7: Cumulative betting return from placing bets with model 1 (red line) and model 2 (green line). The blue line adds cumulatively the spreads of all matches.

	WAIC	p_{WAIC}	WAIC SE
Model 1	9733.2	3.0	73.3
Model 2	9730.5	15.8	73.4

Table 4: The WAIC, p_{WAIC} and the standard error (SE) of the WAIC value for model 1 and 2.

We have seen that both models suffer from overdispersion, meaning that there is not enough variance in the posterior distributions to explain tail events well. The implication of this is that our model, generally speaking, has stronger beliefs in the underdogs than the market. Because the return (loss) was higher than that expected from the spread this could imply that there is a favorite bias in the market and that we would beat the market by always betting on the underdog. This is indeed the case. During 2021, if one were to always bet on the underdog then the average return would be -0.948 (1631 observations). That year has been especially good for betting on underdogs. If we look at the training data (matches between 2017-2020) then the average underdog bet return would be -1.778 (8733 observations). Another observation is that betting on big underdogs (a win chance, based on the bookie, less than 20%) the year 2021 would return +0.476 (301 observations) but on these events our models returned +4.490 (96 observations) and +6.506 (98 observations) for model 1 and 2 respectively. However, we need to be cautious here. Doing the same thing on the training data gives an average underdog return of -2.125 from observations and an average -3.010/-3.188 return from model 1/2. This is surprising because we would expect the model to do better on the data it was trained on. For comparison, when using models 1 and 2 on the whole training data set the prediction accuracy was around 69.5% for both models (higher then the bookie) and the average return was +2.508 and +2.921. This makes one think that a large part of the returns on test data could be due to randomness. To test for this one could shuffle the data so that it is no longer ordered by date and train several new models on different parts of the data while leaving one part out for validation (i.e. cross-validation, see [4, section 1.4]). However, that is beyond the scope of this thesis.

4 Discussion

4.1 Results

In this thesis we fitted to two Bayesian logistic regression models to tennis data. The models were trained on data of tennis games from 2017-2020, and validated on data of tennis games from 2021. Both models made similar predictions of the outcomes of the games and they gave similar results in general. We saw that both models beat random guessing and we also discovered that the cause of this is a favorite bias in the tennis betting market. The reason our models beat random guessing is that they manage to pick up some of this bias in key regions of predictor space, specifically on big underdogs where our models performed very well, comfortably beating the spread. On more even matches our models' performance was in line with random guessing and since these matches are much more numerous the overall return was negative but slightly better than the spread.

It is worth noting that our models beat random guessing but not the "always

bet on the underdog" strategy in terms of returns. Furthermore, since 2021 was an exceptionally good year for underdog betting it is not clear that our model would have performed equally well for a normal year. Before developing the model further one should examine this more closely.

4.2 Possible further improvements

In our posterior predictive checks (PPCs) we saw strong indications of overdispersion. We did not address this in the model at all so a natural first improvement would be to do so. This could be done by assuming that data comes from some other distribution where we have a free variance parameter. A few choices are "probit" and "robit" regression, where where a latent variable of the win probability is assumed to be normally distributed and t-distributed respectively [10].

In our model we used "anonymous" data. That is, we we did not use players directly as predictors, but rather their elos and the bookie odds of games. This means that there is a lot of information that we do not explicitly include in our model, for example a player's skill on a specific tennis court surface. This information is probably baked into bookie odds but it could be beneficial to include it in our model more directly. One way to do that would be to extend the elo to include it. For example by having court specific elo measures as well as an overall elo and include both in the model. We could also customize the K-value of the elo algorithm so that, for example, Grand slam matches are worth more elo. Another improvement to the elo rating could be to adjust elo differently if the game was "close" compared to if it was a total stomp. There are however drawbacks of the elo system that we can not overcome. One such drawback is that it lacks a probabilistic justification. Perhaps it is worth examining different methods of incorporating player specific information, decision trees for ranking could be one such approach. This would have the added benefit of letting us customize the ranking loss function.

Another possible improvement is to take betting market structures into account. Odds is obtained by price discovery and the bookie makes money of the spread between odds. Naturally a bookie must place the opening odds somewhere, which can be shown to be more inefficient than closing odds [13]. To be risk neutral, the bookie will restrict players maximum bet size and move the odds more easily early on, when being more uncertain about the odds. As the total bet turnover increases the odds become more efficient [13], the bookie allows higher bets and is able to lower the spread[4]. What this means for us is that spread might vary between games. We already mentioned that the average test data spread was 2.56% but the lowest and highest observed spread was 1.01% and 4.09% respectively. So, theoretically, a model needs to be right less often on

[4]Not all bookies have enough volume to operate like this. Most just copy the odds from the ones that do, increase the spread and ban players that abuse the inefficiencies created by this strategy.

low spread matches to turn a profit but these matches are likely to have sharper odds, making them harder to beat. In a way, we can think about spread as the bookies' uncertainty. Bookie odds was the by far best predictor so including the information from the spread could result in a model that relies more on other predictors when the spread (uncertainty) is high, perhaps resulting in an overall better model. However, because of the way in which we trained the models I still think removing the spread from bookie probabilities was correct but we could have included the spread in our models as a predictor of its own.

Recall that we assumed in the second model that "bookie prob" and "elo diff" have global effect parameters. From the above discussion about spread that might not be a good assumption. The average spread in the more popular Grand slam matches is 2.35% and then it increases almost linearly for lower series. If we had allowed for "bookie prob" to vary between groups then perhaps the model could have captured some of the uncertainty in the spread (without incorporating a specific variable for it).

References

[1] Gelman, A. Carlin, J. Stern, H. Dunson, D. Vehtari, A. and Rubin, D. *Bayesian Data Analysis*. Third edition. (2013). Chapman & Hall/CRC

[2] Agresti, A. *Categorical Data Analysis*. Third edition. (2016). John Wiley & Sons, Inc., Hoboken, New Jersey

[3] Held, L. and Bové, D. S. *Applied Statistical Inference: Likelihood and Bayes* (2014). Springer-Verlag Berlin Heidelberg.

[4] Bishop, C. M. *Pattern Recognition And Machine Learning*. (2006). Springer Science+Business Media, LLC.

[5] Tennis-Data. http://www.tennis-data.co.uk/data.php. Data last downloaded 2021-12-15.

[6] Santos-Fernandez, E., Wu, P. and Mengersen, K. Bayesian statistics meets sports: a comprehensive review. (2019). *Journal of Quantitative Analysis in Sports*, Vol. 15 (Issue 4), pp. 289-312.

[7] Elo, A. E. *The Rating of Chessplayers, Past and Present*. Second edition. (1978). Arco Publisher, New York.

[8] Edited by: Brooks, S. Gelman, A. Jones, G. and Meng, X. *Handbook of Markov Chain Monte Carlo*. (2011). Chapman & Hall/CRC

[9] Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Keenedy, L., Gabry, J., Bürkner, P. and Modrák, M. (2020). *Bayesian Workflow*.

[10] LIU, C. (2004). Robit regression: A simple robust alternative to logistic and probit regression. In *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*. (A. Gelman and X. L. Meng, eds.) 227–238. Wiley, London. MR2138259

[11] McElreath, R. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Second edition. (2015). CRC Press/Taylor & Francis Group.

[12] Williams, L., V. Liu, C. and Gerrad, H. *How well do Elo-based ratings predict professional tennis matches?*. (2019). NBS Discussion Papers in Economics 2019/03.

[13] Krieger, K. and Fodor, A. Price movements and the prevalence of informed traders: The case of line movement in college basketball. (2013). *Journal of Economics and Business*, Vol. 68, pp. 70-82.

[14] Porter, T. M. probability and statistics. (27 september 2022). *Encyclopedia Britannica*. https://www.britannica.com/science/probability. Accessed 5 January 2023.

[15] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. Equation of state calculations by fast computing machines. (1953). *Journal of Chemical Physics*, 21(6):1087–1092.