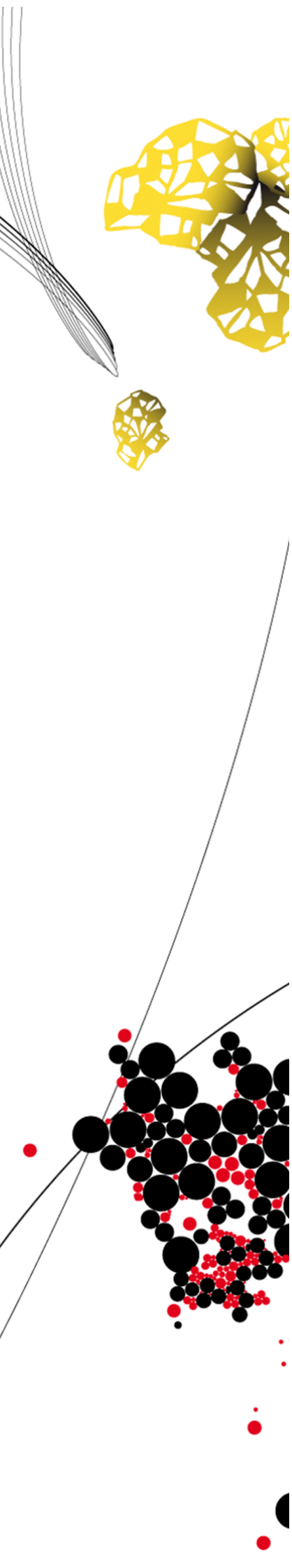

UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

A rating model for individual player qualities based on team results, applied in football

Anatolij I. Babič

MSc Thesis
December 2017

Supervisors:
Prof. Dr. R.J. Boucherie
Dr. J.B. Timmer
Dr. P.K. Mandal
Dr. J. van Haaren

Stochastic Operation Research
Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Preface

Before you lies the master thesis "A rating model for individual player qualities based on team results, applied in football". It has been written to fulfill the graduation requirements of the Applied Mathematics Master's program at the University of Twente, Enschede. The project was undertaken in collaboration with SciSports, a football analytics company that is a spinoff from the University of Twente. From March to December, 2017, I have worked on this project, mostly from the SciSports office in Amersfoort and an improvised home office in Amsterdam. The research was very challenging and I'm happy to say that this work resulted in a novel method for player quality estimation in a multi-player environment.
I would like to thank my supervisors for their contributions and in-depth inspiring discussions regarding this research. Although Richard is not a huge football fan, I'm very happy with his excellent support and advice during this project. I would also like to thank Jan for sharing his expertise regarding the formulation of ideas that resulted in the eventual model.

I hope you will read this thesis with great interest.

Anatoliy Babic

Amsterdam, December 2017

Abstract

There are abundant situations where teams of players compete. The competing players have qualities that influence the outcome of a match, but in some cases, individual contributions are not recorded and only team results can be observed. Examples can be found in sports (football, basketball, volleyball, etc.), e-sports (Dota, StarCraft, League of Legends, etc.), film-making and company management. In this research, we developed an individual player quality inference algorithm which only requires historical team results. Whenever groups of individuals produce collective results and we do not have the data regarding individual contributions, our model can be applied.

Most existing models that are used to estimate individual qualities deal with 1-vs-1 matches, with a single quality per player and a single binary outcome per match. Our model is an extension of existing models; providing a structure to deal with multiple individual qualities, for many-vs-many matches and multiple outcomes per match, each with an ordinal outcome space. The goal was to create a model for any environment where groups of players compete with each other while only the collective results are observed. The research was conducted in partnership with SciSports, a football data analytics company, therefore we chose to apply the model to football.

We considered multiple existing models like ELO, Glicko, Bradley-Terry, Thurstone-Mosteller and the Microsoft TrueSkill. We combined ideas from these models with novel insights to define a probability model, defining the relationship between participating player's qualities and the match-outcome distribution. This relationship is essential for the inference of player quality parameters. The unknown parameters that define the player specific qualities are modeled as latent traits within a latent variable model. We started with a general model, developed in the field of psychometrics, and showed that it is equivalent to our desired model under certain assumptions. Furthermore, we discuss how ordinal observations should be interpreted and we find accurate and useful approximations for the ordinal outcome probability distribution given player participation.

We list several existing estimation methods that can be used to extract estimators from the data by applying our probability model. The methods were taken from other research and modified such that they can be applied to our specific case.

Eventually, we decided to use the Conditional Gaussian Inference method, which has been successfully implemented in Python. We applied the model to a historical football dataset, yielding two qualities per player; attack and defense. The results were tested with a subjective and an objective method, both methods show that our model produces useful results.

Contents

1 Introduction

There are abundant situations in reality where a coalition of players collaborate to achieve a collective set of goals. Such players have certain skills that often cannot be measured directly, but are an explanatory variable for historical and future results. The goal of this research is to find an algorithm that estimates individual player skill quality by using historical results achieved by coalitions of players. We do this by modeling the relationship between quality, performance and realized outcome and applying estimation techniques to infer player quality from historical results.
Some examples of collaborating teams with collective observable outcomes are start-up company entrepreneurs, football players and film-making teams. In film-making, a team of professionals works together to produce a profitable and qualitative end product. In a start-up company, a small team of entrepreneurs and professionals collaborate to build a profitable business. In the game of football two teams compete with the objective to score more often than their opponent. In these environments, it is possible to observe collective results, but there is a need to understand contributions of individuals to the achievements of a team. Individual achievements are often difficult to extract because of the collaborative nature of the environment. Most *individual results* are (partly) a team performance, rather than purely an individual performance.

1.1 Player quality estimation

Knowledge regarding player qualities can be very useful for decision-making purposes. Numerical evaluations of the qualities of film-makers can be used in the decision-making progress of funding allocation for future films (Timmer *et al.* , 2017). Whenever playing online on the Xbox, the opponents are done by a match-making system that uses player quality estimates to pair evenly matched players, hereby avoiding that an advanced player will play against players that are new to the game (Herbrich *et al.* , 2007). Accurate player quality estimates provide an understanding of hidden variables that can be used to explain historical and predict future performances.
In game theory, the importance of individual contributions to coalitions has been extensively researched. Coalitions have a value, which can be distributed over the coalition members in according to certain criteria. The Shapley value is a unique *value distribution* that follows from a set of desirable properties. The Core is a set of value distributions that cannot be improved upon by sub-coalitions (Gillies, 1959).
A different approach to estimating player quality, is by benchmarking a player's strategy to *the optimal strategy*. Another approach would be to perform cognitive or physical tests and use the results as a proxy for player quality.
Our research focused on environments where we *do not* have observations of individual performances of players. Our approach is data-driven; the only input required is data of match outcomes and player participation. Our model requires a very limited amount of domain knowledge; i.e. what we require from domain experts are weights that assign importance to qualities in certain situations. We *do not* need to model the game environment, understand successful strategies or know the exact rules.

1.2 Existing models

There are existing models that assign a value to player qualities.
In game theory, the value of individual contributions in coalition games is a very important result. Whenever the values of all sub-coalitions are known the value of individuals within a game can be characterized by the Shapley value (Shapley, 1953).
In some environments, we can have a round-robin tournament (all-play-all) schedule and yield a ranking for all the players. The main disadvantage of this is the large amount of (possibly irrelevant) matches that need to be played. A different approach is to keep track of player qualities in a so-called *rating system*. A rating system keeps track of player ratings, can predict the outcome probability from the ratings of player and updates the ratings after every encounter. Ideally, such a rating is a representation of skill.
The first rating system was the ELO-rating (Elo, 1978), developed by Árpád Élő for the game of chess and is used to determine the official chess (FIDE) world ranking. The system can be used to objectively calculate the relative levels of skill in a competitor-versus-competitor environment, providing insight into player quality for games where round-robin schedules are infeasible. The ELO-rating system has been applied to other games like Scrabble, Football, American Football and Go. These extensions of the ELO-rating calculate only the quality of teams (coalitions), rather than the quality of individual players. The Glicko (Glickman, 1999) model is an extension of the ELO

method. It models the player quality as a Gaussian random variable. This improved the error and convergence of the ratings. The Microsoft TrueSkill algorithm (Herbrich *et al.* , 2007) model is designed for multi-player environments, applying state-of-the-art modeling techniques (factor graphs) and inference algorithms (Expectation Propagation).
Player qualities can also be estimated with statistical approaches. The plus-minus method applies linear regression between player participation and outcomes (Fitzpatrick, 2017; Rosenbaum, 2004). The qualities of film-makers can be estimated by a linear estimator using the success of films made in the past (Timmer *et al.* , 2017).
Another existing individual player rating model has been developed by SciSports, the company for which we performed this research. The model is called SciSkill, was developed especially for football and covers more than 70.000 football players from the whole world. It is a difference based approach inspired by the ELO rating system. The algorithm contains a lot of components that solve problems in a very pragmatic way. The model was built with a focus on application, therefore certain parts of the algorithm lack scientific justification. The model is not published, and therefore we will not discuss it in the literature study.

1.3 Our approach

Our focus is finding a method to estimate player qualities, without using any individual data, only team results. We assume that player quality, an unobservable variable, has a stochastic relationship with historical outcomes and future outcomes. The historical outcomes can be used to estimate the qualities, and the estimates we yield can be used for prediction of future matches. We assume the player quality to be non-deterministic, therefore it is modeled as a random variable. As players perform in teams, against other teams, we require a way to aggregate performances. We assume that a useful aggregation can be achieved by a weighted sum of the individual qualities. This aggregation has a non-linear relationship with the observed outcomes. We define this relationship with a probability model.
After the construction of the probability model, we apply estimation techniques that extract player quality parameter from the historical data. The estimation techniques use different assumptions and optimization criteria. The estimators we find are tested according to a subjective criterion (player rankings) and an objective criterion (future match prediction).
Due to the assumptions and modeling choices, the abstract problem we solve in this research is parameter estimation of a normal distributions, if we only observe a specific non-linear transformation of an affine transformation of the realizations of this normal distribution.

1.4 Contributions

The rating model we developed in this thesis is an extension and improvement of several existing rating models. The main difference is that most models focus on 1-vs-1 matches with only a single quality per player. In environments where players play in teams, such models aggregate player ratings to a single rating per team. The first contribution of our model is that it can deal with many-versus-many matches, not focusing on the coalition strengths but on the individual player strengths. Our second contribution is that our model can estimate multiple qualities for each player player. This allows differentiation between players that have different qualities and roles. The third contribution of our model is that it is built for an environment with an ordinal outcome space. Most other models focus on a binary (sometimes extended to a ternary) outcome space. Effectively our inputs contain more information and therefore should produce more accurate estimates. Due to the fact our model utilizes an ordinal outcome space, it naturally deals with winning margin. Our fourth contribution is that our method not only yields point estimates for player qualities but also player quality uncertainty and covariance between our estimates of player qualities. The final contribution of our work is that it can predict the outcome distribution for specific coalition comparison outcomes. The quality estimates we yield from historical data are used as input, and can be used to predict the probability distribution of the outcome of future matches. Such predictions can be used to validate our methodology.

2 Literature study

There have been efforts to create effective algorithms that infer the qualities of individuals. Such qualities are latent variables, as they cannot be observed directly, but have a relationship with observable quantities. Therefore accurate quality models can be used to analyze historical outcomes and predict future outcomes.
The first type of quality estimation method we discuss is strategy benchmarking. The idea of this method is to assess the quality of players, by benchmarking their strategy to the optimal strategy. We discuss this approach in Subsection 2.1.
There are other methods that generate statistical metrics to assess player qualities. Such metrics are based on smart counting of past performances, which often misses out on a lot of context information. We discuss some of these statistical metric methods in Subsection 2.2.
Rating systems include a very important contextual element; the quality of the opponent is taken into account. Rating systems produce estimates of player ratings (unobservable latent variables), that are a representation of player quality. Rating systems use historical data to infer estimates of player ratings. Some possible data sources are polls, betting odds or historical results. A very important building block of a rating system is quality inference based on historical comparisons of two objects. Such experiments are called pairwise comparisons, and we describe methods that use these in Subsection 2.3. The outcomes of pairwise comparisons are traditionally binary, but in reality, outcomes often are ordinal or continuous. The most famous pairwise comparison models are the Thurstone-Mosteller and Bradley-Terry models, discussed in 2.3.2 and 2.3.1. We will discuss some extensions of the Bradley-Terry model that allows for draws in Subsection 2.4. We will continue with discussing rating systems in Section 2.5. The most famous rating systems are ELO and Glicko, these will be elaborated in 2.5.1 and 2.5.2 respectively.
A relatively new approach is the TrueSkill$^{\text{tm}}$ developed by Microsoft Research, specifically to achieve fair matchmaking for online games on the Xbox, Microsoft's online gaming platform. The TrueSkill methodology applies Bayesian graphical modeling to infer player rating distributions from past results, we discuss it separately in Subsection 2.6. Lastly, we discuss a coalition assessment model, that has been applied successfully to estimate the qualities of film-makers in Subsection 2.7.
Throughout this section, we will require the definition of likelihood $\mathcal{L}$ of parameters given a set of observations. Whenever we have parameters θ and data D, we define the likelihood of the estimator $\hat{\theta}$ of the parameters θ as:

$$L\left(\theta = \hat{\theta}; D = d\right) = P_{\hat{\theta}}\left(D = d\right) \tag{1}$$

Here we use the notation $P_{\hat{\theta}}\left(D = d\right)$, which is the probability that $D = d$ under the condition that the parameter $\theta = \hat{\theta}$. The estimator of θ that maximizes the likelihood, is defined as the Maximum Likelihood Estimator (MLE):

$$\hat{\theta}^{MLE} = \underset{\hat{\theta}}{\operatorname{argmax}}\, P_{\hat{\theta}}\left(D = d\right) \tag{2}$$

$$= \underset{\hat{\theta}}{\operatorname{argmax}}\, H\left(P_{\hat{\theta}}\left(D = d\right)\right) \tag{3}$$

Here $H : [0, 1] \to \mathbb{R}$ must be a strictly increasing function.

2.1 Strategy benchmarking

The idea of this approach is to perform player quality estimation by evaluating the strategy (all decisions and actions) of a player with respect to *the game theoretical optimal strategy*. Finding such an optimal strategy is very complex, but for our purposes, a strategy that can beat top human players is enough and evaluate game-states. The main assumption is that players with a high quality have a superior strategy over players with a high quality.
In general, it is very difficult for an algorithm to determine a good strategy in a game. In real-world scenarios, there are intractably many possible tactics and strategies. In game environments successful strategies are often not fully understood, the outcomes are influenced by unobservable stochastic variables, and interactions between players are very unpredictable. An individual game like chess is finite, discrete, non-stochastic and still has enormous complexity. Researchers required a very long time to make an algorithm that can beat the best human player (King, 1997). Once a computer algorithm is able to beat top human players in a certain game, we refer to it as *solved*. The idea is, that the strategy of such an algorithm can be used as a proxy for the *optimal strategy*. Algorithms based

on reinforcement learning techniques explicitly have an internal game-state evaluation model, which makes it possible to evaluate all the decisions and actions given a certain game-state. This evaluation could be performed on past performance of individuals. A good example of such a method is the *centipawn loss* system developed in chess, which calculates how much centipawns ($\frac{1}{100}$ of a pawn) a player loses on average compared to *the computer move*. A low centipawn loss can be seen as a good proxy for player quality. A nice feature of the centipawn methodology is that it has a dimension (centipawns lost per move), and therefore can be interpreted intuitively.
Due to increased computing power and the utilization of machine learning techniques, progress in the field of *solving* games has been: examples are Jeopardy (general knowledge quiz), Go (deterministic board game) and No-Limit Texas Hold'em (stochastic card game).

Remark 1. *Some games are not solved due to the fact that not only decisions need to be made, but they also need to be executed with precision. Examples of such games are football, golf, ping-pong, Dota 2 and StarCraft. In football, Robocup is an ongoing initiative, started in 1997 (Robocup, 2017), with the ambitious goal of designing humanoid robots to beat real humans at football before the year 2050 (Kitano* et al. *, 1997). There has been research with the goal of designing robots that can autonomously play ping-pong (Peters* et al. *, 2013). Very recently, the company OpenAI successfully developed a bot for 1-vs-1 matches in the game Dota 2, beating top human players (OpenAI, 2017).*

We expect more games to be solved by algorithms in the future. We expect rating systems based on the described approach to be very accurate. An assumption of this approach is that an optimal strategy in human vs human matches is equal (or at least similar) to the strategy a computer chooses. Unfortunately, this is not always true; psychological mind-games and intimidation can play a big role in human-vs-human matches, while a computer algorithm would never be influenced by this. An important *quality* in human-vs-human matches is to understand your opponents (weaknesses and strengths), while this *quality* is irrelevant against a strictly better computer algorithm and therefore will not be measured.

2.2 Statistical player metrics

There are models that focus on finding statistical metrics to quantify player performance. In the sport of football examples are: goals scored, successful pass percentage and expected goals. All these metrics are a weighted counting technique, where often context is not fully captured. Goals scored can, for example, be skewed because a player played a lot, or because a player takes penalties. It is always interesting to normalize quantities (apply relevant dimensions), e.g. non-penalty goals per 90 minutes played instead of total goals. A nice property of such methods is that statistical assessments (metrics) have a dimension, which often allows for intuitive interpretation and usage. The plus-minus statistic is a more elaborate statistical method, discussed in detail in Section 2.2.1.
In (Tiedemann *et al.* , 2011) the performance of players is evaluated based on a non-parametric concave meta-frontier approach. The meta-frontier defines a theoretical optimal player performance, based on the playing time and position of the *best player*. This permits estimation of all players' efficiency. A positive correlation has been found between players' efficiency and their historical team performance. We believe the main reason for this relationship is that (within this method) goals for and goals against form a very important factor in determining the efficiency of players and success of teams. The method does not provide any predictive capabilities.
A data-driven method to determine the ability of soccer players entirely based on the value of their completed passes was developed by (Brooks *et al.* , 2016). Passes are valued according to location and shot opportunities generated. The relationships are learned from data and mostly work for offensively minded players.

2.2.1 Plus-minus statistic in basketball and ice-hockey

The plus-minus statistic (PM) is a statistical measure to determine the average added value of basketball players in the NBA (Rosenbaum, 2004) and NHL (Fitzpatrick, 2017). The simplest interpretation of this rating system is a virtual counter of the total goals a team scores minus the total goals a team concedes, whenever a player is in the field. The system looks at player participation, and sets up a linear system for each part of the match where there is no substitution:

$$M = q_{home} + \delta_0 q_0 + \delta_1 q_1 + \delta_2 q_2 + ... + \delta_k q_k + \epsilon \tag{4}$$

The variable M represents the difference in average points per possession, q_{home} is a variable that accounts for home advantage, q_i is the quality of player i, for $i = \{1, 2, ..., k\}$, and ϵ is an error term.

Only players that reach a minimum amount of minutes receive a personal rating, other players are grouped in the variable q_0. Lastly; $\delta_i = 1$, if player i plays for the home team, and $\delta_i = -1$ if a player plays for the away team. The plus-minus statistic is very dependent on the context wherein a player performs, mostly the quality of his own team and the opponents team. Because of the lack of context incorporated in the plus-minus calculation is biased and a player with the exact same skill can get a different rating.
The plus-minus statistic can be extended to the adjusted plus-minus (APM) statistic, by separating the single equation for each match into two equations. This separation is done on a player level; yielding an equation for the time a player was in the field, and the other equation for the time this player was not in the field. The performance (realized score margin) of a team during the part of the match when the player is in the field and when he is on the bench are compared. This way we can detect differences in performance of a team whenever a single player is playing. By using this approach we eliminate the influence of own team strength and opponent strength.
One of the main requirements for using PM is a high scoring frequency. Furthermore, the APM approach works optimally in sports where a line-up is constantly changing during the game.

2.3 Pairwise comparisons

In this section, we will call the players/objects that are compared p_i, and they will have a quality rating of q_i. The rating of all players is represented in the vector q. Depending on the model, player qualities are defined as a parameter or a random variable. In the case that the qualities are random variables, we take $q_i \sim Q_i$. We call the variable D_{ij} the outcome of the pairwise comparison of i and j and define it such that:

$$D_{ij} = \begin{cases} 1 & \text{if } i \text{ wins} \\ 0 & \text{if } j \text{ wins} \\ \frac{1}{2} & i \text{ draws } j \end{cases} \tag{5}$$

It holds that $D_{ij} + D_{ji} = 1$. In general, i and j could be compared multiple times, but we do not account for this in our notation.
There has been some fundamental research into pairwise comparisons. Two of the most used models in literature are the Bradley-Terry (Bradley & Terry, 1952) and Thurstone-Mosteller (Thurstone, 1927), (Mosteller, 1951) rating systems. Both approaches are not developed explicitly to deal with draws or multiple players, but the models can be extended to allow for such cases. We will discuss some examples of extensions in Section 2.4.

2.3.1 Bradley-Terry model

The Bradley-Terry approach (Bradley & Terry, 1952) considers that all the objects which are being compared have a constant rating parameter. In the original approach pairwise comparison experiments are considered with a binary outcome, not allowing for draws, with probabilities defined as:

$$P_q(D_{ij} = 1) = \frac{q_1}{q_1 + q_2} = 1 - P_q(D_{ij} = 0) \tag{6}$$

Parameters of teams can be estimated efficiently, multiple methods have been developed to achieve this. One method is a recursive Minorization-Maximization procedure (Hunter, 2004) applied to the log-likelihood function of observations. We define all our results as D, and calculate the probability of observing the results d, we can extract w_{ij} as the number of times i has beaten j to get:

$$P_q(D = d) = \prod_{i,j} \left(\frac{q_i}{q_i + q_j} \right)^{w_{ij}} \tag{7}$$

$$\log P_q(D = d) = \sum_{i,j} w_{ij} \log(q_i) - w_{ij} \log(q_i + q_j) \tag{8}$$

The maximum likelihood estimators for the parameters can be found with a method similar to logistic regression. A feasible solution exists under the condition that there is no partition of the players in two groups, where the outcomes of all comparisons between players from different groups have one-sided outcomes. If such a partition, of the whole player set P, were to exists, say p and p^c so that $p \cup p^c = P$ and $\forall_{i,j} : i \in p \wedge j \in p^c \implies w_{ij} \geq 0 \wedge w_{ji} = 0$. In essence; no player from p^C has ever beaten a player from p. To avoid the trivial case (no games at all were played between players

from p and p^C) we require $\exists_{i,j} i \in p \wedge j \in p^c$ $(w_{ij} > 0)$. Under the previous scenario, our maximum likelihood solution will become:

$$\hat{q}^{BT-MLE} = \underset{q}{\operatorname{argmax}}(\log P_q(D = d)) \tag{9}$$

$$\hat{q}_i^{BT-MLE} \to \infty \qquad \forall_i, i \in p \tag{10}$$

$$\hat{q}_j^{BT-MLE} \to -\infty \qquad \forall_j, j \in p^C \tag{11}$$

The values we calculate will diverge, which is a useless answer for our problem. This problem may occur when we have a small dataset and results between two groups are very one-sided, but fortunately it can be avoided by regularizing the player rating parameters. This can be done, according to a Bayesian approach by taking a prior distribution over the ratings $q \sim Q$. We are left with the following log-likelihood to be maximized:

$$p_{D,Q}(d, q) = P(D = d|Q = q)p_Q(q) \tag{12}$$

$$\log(p_{D,Q}(d, q)) = \log(P(D = d|Q = q)) + \log(p_Q(q)) \tag{13}$$

$$\hat{q}^{BBT-MLE} = \underset{q}{\operatorname{argmax}} \left[\log(P(D = d|Q = q)) + \log(p_Q(q))\right] \tag{14}$$

2.3.2 Thurstone-Mosteller model

The Thurstone-Mosteller model (Thurstone, 1927) assumes that player ratings are random variables with a normal distribution, thus $q_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. In general, σ_i^2 is player dependent, but often for simplicity it is chosen the same for all players. The model now states that the winner of a paired comparison is the player with the highest player performance, which has the same distribution as his rating. We get the following:

$$\begin{aligned}
P\left(D_{ij} = 1|q_i \sim \mathcal{N}(\mu_i, \sigma_i^2), q_j \sim \mathcal{N}(\mu_j, \sigma_j^2)\right) &= P\left(q_i > q_j|q_i \sim \mathcal{N}(\mu_i, \sigma_i^2), q_j \sim \mathcal{N}(\mu_j, \sigma_j^2)\right) \\
&= P\left(X > 0|X \sim \mathcal{N}(\mu_i - \mu_j, \sigma_i^2 + \sigma_j^2)\right) \\
&= 1 - P\left(X < 0|X \sim \mathcal{N}(\mu_i - \mu_j, \sigma_i^2 + \sigma_j^2)\right) \\
&= 1 - \Phi\left(\frac{\mu_j - \mu_i}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right) \\
&= \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right)
\end{aligned}$$

Here we use that Φ is the standard normal cumulative distribution function. This shows that the Thurstone-Mosteller model is a linear probit model (Albert & Chib, 1993). The ratings can be estimated efficiently by assuming that the rating distributions are constant during the complete period during which the pairwise comparisons were performed. The complete likelihood function becomes:

$$q \sim \mathcal{N}(\mu, \Sigma) \tag{15}$$

$$\mathcal{L}(\mu, \Sigma; D = d) = P_{\mu,\Sigma}(D = d) \tag{16}$$

$$= \prod_{i,j} \left[D_{ij} \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right) + (1 - D_{ij}) \left(1 - \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right)\right)\right] \tag{17}$$

$$\hat{\mu}^{TM-MLE}, \hat{\Sigma}^{TM-MLE} = \underset{\mu,\Sigma}{\operatorname{argmax}} \left[\mathcal{L}(\mu, \Sigma; D = d)\right] \tag{18}$$

Gibbs sampling can be used to efficiently find the MLE for the above likelihood expression.

2.4 Pairwise comparison models with non-binary outcomes

In a lot of cases, pairwise comparisons have a non-binary outcome space. For example in the sports of hockey, football, and chess matches can all end in a draw. One solution is proposed by the ELO rating model discussed in 2.5.1, by looking at the expected outcome. In this section, we will look at

extensions of the Bradley-Terry model with an outcome space of N possibilities.
A generalized Bradley-Terry model could look as follows;

$$P(Y_{ij} = k) = \frac{e^{z_k(i,j)}}{\sum_{n=1}^{N} e^{z_n(i,j)}} \tag{19}$$

Here the function z_n is specific for outcome n, but depends on the participants i and j. An extension of the Bradley-Terry approach by for the specific case of three outcomes, where two are decisive and one is non-decisive, a model was proposed by (Davidson, 1970):

$$P\left(Y_{ij} = 1\right) = \frac{e^{q_i}}{e^{q_i} + e^{q_j} + e^{\lambda + \frac{1}{2}q_i + \frac{1}{2}q_j}} \tag{20}$$

$$P\left(Y_{ij} = \frac{1}{2}\right) = \frac{e^{\lambda + \frac{1}{2}q_i + \frac{1}{2}q_j}}{e^{q_i} + e^{q_j} + e^{\lambda + \frac{1}{2}q_i + \frac{1}{2}q_j}} \tag{21}$$

$$P\left(Y_{ij} = 0\right) = \frac{e^{q_j}}{e^{q_i} + e^{q_j} + e^{\lambda + \frac{1}{2}q_i + \frac{1}{2}q_j}} \tag{22}$$

Another solution discussed in the same paper, the Rao-Kupper tie model, gives the following equations:

$$P(Y_{ij} = 1) = \frac{e^{q_i}}{e^{q_i} + \lambda e^{q_j}} \tag{23}$$

$$P(Y_{ij} = \frac{1}{2}) = (\lambda^2 - 1)\frac{e^{q_i + q_j}}{(e^{q_i} + \lambda e^{q_j})(e^{q_j} + \lambda e^{q_i})} \tag{24}$$

$$P(Y_{ij} = 0) = \frac{e^{q_j}}{\lambda e^{q_i} + e^{q_j}} \tag{25}$$

where we require that $\lambda \geq 1$. In the case that $\lambda = 1$ reduces to the standard Bradley-Terry model.
These two examples are both valid varieties of the generalized model described in Equation (19). We conclude that there is no single choice for the functions $z_n(i, j)$, and for specific applications, tailor-made solutions should be developed.

2.5 Rating models

In this subsection, we will discuss two rating models that are often used; ELO and Glicko. These models, as most rating models, use a latent variable that is a representation of player quality, and a mapping from the qualities of all players in a match to the match outcome. The main difference between the models is that ELO only estimates the first moment of player ratings, while Glicko also estimates the second moment. The probability model that is necessary for rating inference, can be used to predict future fixtures.
The main difference between previously discussed models and rating models, is that rating models incrementally process historical data. Therefore the produced ratings are a time series, showing the development of the rating rather than just a point estimate. Another attractive feature of rating systems is that they allow a continuous outcome space, as rating updates are performed as a function of difference between performance and expected performance. We use the following notation; q_i is the prior and q_i^{new} is the posterior rating of player i. We define dS_{ij} as the prior rating difference and dS_{ij}^O as the observed strength difference for a match between player i and j. Also, we need a monotonically increasing function $g(\cdot)$ that gives us the update magnitude based on observed rating difference.
We get the following equations for the update after a match between player i and player j:

$$dS_{ij} = q_i - q_j = -dS_{ji} \tag{26}$$

$$q_i^{\text{new}} = q_i + g(dS_{ij}^O - dS_{ij}) \tag{27}$$

$$q_j^{\text{new}} = q_j + g(dS_{ji}^O - dS_{ji}) \tag{28}$$

It makes sense to choose g such that we have $g(0) = 0$ and $g(-x) = g(x)$, so that over-performance increases and under-performance decreases ratings, and the total amount of points in the system remains constant.

2.5.1 ELO-rating

The ELO-rating is a widely used rating system, initially developed to determine the relative strengths of chess players, by applying an iterative inference process on game outcomes (Elo, 1978). The relative strengths are parametrized as ratings, and can be used to generate match outcome probabilities for players that have never played each other.
The approach uses the Bradley-Terry paired comparison formula to determine match outcome probabilities. It applies a logarithmic transform on the ratings, using $\tilde{q}_i = e^{q_i}$ where $\tilde{q}_i$ are the Bradley-Terry strength parameters. We get the following relationship:

$$E_{ij} = E[Y_{ij}] \tag{29}$$
$$= \frac{\tilde{q}_i}{\tilde{q}_i + \tilde{q}_j} \tag{30}$$
$$= \frac{e^{q_i}}{e^{q_i} + e^{q_j}} \tag{31}$$
$$= \frac{1}{1 + e^{q_j - q_i}} \tag{32}$$

Where E_{ij} corresponds to the average amount of points player i gets when competing against player j, where a win counts for one point, draw counts as $\frac{1}{2}$ point and a loss as zero points. We can observe that this implies that match outcome probabilities follow directly from the rating difference, $q_j - q_i$. The updating of ratings given a match outcome is done relative to the expected outcome, which is calculated with the player ratings. Whenever a player over-performs(under-performs) his rating will become higher(lower). This way the ratings slowly converge to their real value. In the ELO model the update equations look as follows:

$$q_i^{\text{new}} = q_i + K \cdot (Y_i - E_i) \tag{33}$$

Where Y_i is the total points and E_i is the expected amount of points of player i in the period since the last rating change. K is a parameter that determines the magnitude of the rating change. K is positive such that ratings of players increase (decrease) whenever players overperform (underperform). In general, K should be chosen higher for important matches. Friendly matches should have a lower K factor, than a world cup final match. The choice of the K remains a domain and match specific problem. The system has received a lot of theoretical critique, and statistical improvements have been proposed by Glicko-model, discussed in the next subsection. Nonetheless the ELO-model remains the standard in a lot of disciplines. The main reason is that the ELO-model is much easier to understand, explain and implement than any other available alternative.

2.5.2 Glicko-rating

The Glicko-rating system (Glickman, 1999) was developed by Mark Glickman, it is an extension to the ELO-rating but it specifies the rating as a random variable. The rating of player i has a normal distribution, we use the following notation:

$$Q_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{34}$$
$$p_{Q_i}(q_i) = \mathcal{N}(q_i; \mu_i, \sigma_i^2) \tag{35}$$

The method applies an incremental approximate Bayesian estimation procedure to infer the player's skill distribution from past results. The model defines an initial prior rating distribution and performs updates based on the outcomes of games. These updates occur in batches, by assuming that the player's posterior distribution can be determined by integrating out the opponents strength parameter over their prior distribution. In the following equations, we will not index the parameters of the player under consideration, i.e. the player whom posterior parameters we are estimating. We use $q_i \sim Q_k$ as

the strength of the player under consideration and $q_k \sim Q_k$ as the prior strength of his k^{th} opponent.

$$f_Q(q_i|D=d) = \int ... \int f_{Q_i}(q_i|Q_1=q_1,...,Q_N=q_N,D=d)\mathcal{N}(q_1;\mu_1,\sigma_n^2)...\mathcal{N}(q_N;\mu_n,\sigma_n^2)dq_1...dq_N \tag{36}$$

$$\propto \int ... \int \mathcal{N}(q_i;\mu_i,\sigma_i^2)\mathcal{L}(Q_i=q_i,Q_1=q_1,...,Q_N=q_N;D=d)\mathcal{N}(q_1;\mu_1,\sigma_1^2)...\mathcal{N}(q_N;\mu_N,\sigma_N^2)dq_1...dq_N \tag{37}$$

$$= \mathcal{N}(q_i;\mu_i,\sigma_i^2)\prod_{j=1}^{N}\int \mathcal{L}(Q=q,Q_j=q_j;D_{i,j}=d_{i,j})\mathcal{N}(q_j;\mu_j,\sigma_j^2)dq_j \tag{38}$$

$$\propto \mathcal{N}(q_i;\mu_i,\sigma_i^2)\prod_{j=1}^{N}\int P(D_{i,j}=d_{i,j}|Q=q,Q_j=q_j)\mathcal{N}(q_j;\mu_j,\sigma_j^2)dq_j \tag{39}$$

Here we use that $\mathcal{N}(x;\mu,\sigma^2)$ is the probability density function of a normal distribution with mean μ and variance σ^2, evaluated in the point x. In Equation (39) we use the $D_j \in \{0,1\}$, which is the subset of D with the relevant data of outcomes of matches between player j and the player under consideration.
To proceed we need to have the outcome probability given the player ratings. Just like with ELO, as shown in equation (32), this is taken as a logistic distribution. Furthermore, the author uses an approximation from (Crooks, 2013) for the integral in Equation (39):

$$P(D_{i,j}=Y_{i,j}|Q_i=q_i,Q_j=q_j) = \frac{\left(e^{q_i-q_j}\right)^{Y_ij}}{1+e^{q_i-q_j}} \tag{40}$$

$$\int P(D_{i,j}=Y_{i,j}|Q_i=q_i,Q_j=q_j)\mathcal{N}(q_j;\mu_j,\sigma_j^2)dq_j = \int \frac{\left(e^{(q_i-q_j)}\right)^{Y_{ij}}}{1+e^{(q_i-q_j)}}\mathcal{N}(q_j;\mu_j,\sigma_j^2)dq_j \tag{41}$$

$$\approx \frac{\left(e^{g(\sigma_j^2)(q_i-q_j)}\right)^{Y_{ij}}}{1+e^{g(\sigma_j^2)(q_i-q_j)}} \tag{42}$$

$$g(\sigma_j^2) = \frac{1}{\sqrt{1+\frac{3\sigma_j^2}{q^2}}} \tag{43}$$

Using this approximation, we yield the update equations for the rating of player i:

$$\mu_i^{new} = \mu_i + (\frac{1}{\sigma_i^2}+\frac{1}{\delta_i})^{-1}\sum_{\substack{j=1\\j\neq i}}^{N} g(\sigma_j^2)(Y_{ij}-E(Y_{ij}|\mu_i,\mu_j,\sigma_j^2))$$

$$\sigma_i^{new} = (\frac{1}{\sigma_i^2}+\frac{1}{\delta_i^2})^{-\frac{1}{2}}$$

$$E[Y_{ij}|\mu_i,\mu_j,\sigma_j^2] = \frac{1}{1+e^{-g(\sigma_j^2)(\mu_i-\mu_j)}}$$

$$\delta_i^2 = \left[\sum_{\substack{j=1\\j\neq i}}^{N} g(\sigma_j^2)E\left[Y_{ij}\middle|\mu_i,\mu_j,\sigma_j^2\right]\left(1-E\left[Y_{ij}\middle|\mu_i,\mu_j,\sigma_j^2\right]\right)\right]^{-1}$$

The Glicko-system also describes a backward filtering step, using a Kalman Filter approach to extract improved estimators for previous time steps.

2.5.3 Dutch tennis league rating system

The Dutch tennis federation (KNTB) uses a player rating system call DSS that is a Dutch acronym for "Dynamic Playing Strength" (KNTB, 2017). A conceptual difference with most other ranking models is that a lower score is considered favorable. The model can be seen as an estimator of playing

strength and uses the following formulae:

$$R_i^{(k)} = \begin{cases} q_j - 1 & \text{if } Y_{ij} = 1 \text{ and } q_i > q_j - 1 \\ removed & \text{if } Y_{ij} = 1 \text{ and } q_i < q_j - 1 \\ q_j + 1 & \text{if } Y_{ij} = 0 \text{ and } q_i < q_j + 1 \\ removed & \text{if } Y_{ij} = 0 \text{ and } q_i > q_j + 1 \end{cases} \tag{44}$$

$$q_i^{new} = \frac{1}{|M|} \sum_{m \in M} R_i^{(m)} \tag{45}$$

$$M = \{R_i^{(k)} | R_i^{(k)} \neq removed\} \tag{46}$$

Here we have that M is a collection of matches that are not "*removed*" that were played during a certain year. During the start of each year, last year's results are used to form the new player rating. The driving idea behind this rating system is that players are expected to win against an opponent with a rating that is one point higher (worse). Every player keeps a record of scored points; the score is relative to the opponent. Winning from a player means that your rating should be one less than his current rating, therefore your result record will contain this score. For players with large differences, more than one rating point, the expected outcome (lower rated player wins) will be classified as *removed*. This means that a theoretical property of this rating is that players cannot improve their rating whenever they play against much lower rated players, but can worsen their rating (drastically). For this reason, it is very unattractive for competitive players that focus on getting a low rating to play against much weaker players.
There are manual, non-mathematical, adjustments to the model to deal with new players, player inactivity and infrequent playing results. The documentation explaining the system even contains a subsection elaborating that in some (extreme) cases the federation can manually adjust the ratings of players if there is a reason to do so. This shows that the federation employs the rating system as a guideline; adjusting where needed to ensure correctness in extreme cases.

2.6 Microsoft TrueSkill

The TrueSkill rating system (Herbrich *et al.* , 2007) has been developed by Microsoft to be used in a broad spectrum of games offered on their Xbox game console. The main goal of this rating system is to match players with equivalent skills, in essence maximizing draw probability of matches. This way competitive matches between users on the platform ideally are between players of equal strength. TrueSkill is also suitable for multi-player games; it assumes that player qualities are additive. In the following sections, we will explain the model and discuss its performance.

2.6.1 Model

The model uses factor graphs to create a complete probability distribution of individual player skills, team skills, and game outcomes. The skill of individual i is modeled by the random variable $M_i \sim \mathcal{N}(\mu_{M_i}, \sigma^2_{M_i})$. Whenever a player performs, his performance is $q_i \sim Q_i = \mathcal{N}(M_i, \beta)$, where β is a constant performance uncertainty for all players. Players perform in teams, and the team performances to be additive, thus:

$$T_j = \sum_{i \in A_j} Q_i = \mathcal{N}(\sum_{i \in A_j} M_i,\ \beta \cdot N_{A_j}) \tag{47}$$

$$N_{A_j} = |\{i \in A_j\}| \tag{48}$$

Here A_j is the set of indices of players in coalition j. The eventual ranking of the teams within a match is considered to be a direct consequence of the team performances, we define such ranking as $r = \{r(1), .., r(n)\}$ such that $i \geq j \implies t_{r(i)} \leq t_{r(j)}$ where $t_k \sim T_k$ and $r(m)$ is the index of the team that is ranked in the m_{th} place. A lower ranking is considered to be a better. The model introduces dummy variables that indicate the differences between the performance of adjacent teams represented by $d_k = t_{r(k)} - t_{r(k+1)}$. Some teams have the same rank, this happens whenever $d_k < \epsilon$. The dummy variables are only considered for adjacently ranked teams. The idea is to represent this model in a factor graph, yielding an efficient representation of joint probability distribution. A factor graph is a bipartite graph containing all factors of the probability distribution and can be a very efficient way to define a complex probability distribution. Because the graph contains no cycles an efficient algorithm, the sum-product algorithm can be applied to perform inference. Historical results can be

fed to the algorithm to produce improved beliefs of player quality parameters.
A very important part of the algorithm is the approximation method Expectation Propagation algorithm (EP) by Tom Minka, one of the authors of the TrueSkill model. EP is a method to approximately factorize a probability distribution iteratively, by optimizing single factors during each iteration.
A team's skill is set equal to the sum of individual participant skills. Some mathematical techniques like message passing, belief propagation, and expectation propagation, are used for graph inference using game outcome information. Some of these techniques are only exact for normally distributed variables. Therefore within the algorithm, the distribution of non-Gaussian variables is approximated by Gaussians. The approximation is done by minimizing Kullback-Leibler divergence, which with an approximation by a Gaussian distribution comes down to moment matching (Ranganathan, 2004).

2.6.2 Model performance

The model performs very well; rating convergence and outcome prediction are better than for comparable models. The model is twice as inefficient as the theoretical limit (MacKay, 2002, Shannon entropy). Throughout the model, several assumptions are made, we summarize them in the following list:

- The team performance is the sum of individual team member performances, it can be seen as the L^1-norm of the vector with player qualities. Other research papers have found that the inference algorithms can be modified to work with a weighted average of the player qualities in a different norm than the L^1-norm (Nikolenko & Sirotkin, 2011). By using an L^n-norm with $n > 1$ ($n < 1$), we can achieve the behavior that exceptionally good players have larger (smaller) influence.
- The outcome of a match is modeled as a ranking of teams, allowing for draws but not for margin of victory" into account. This could be achieved by defining multiple parameters like ϵ, that would enforce distances between team performances given a specific margin.
- Inference is performed purely based on the final ranking of the teams. This means that margin of victory cannot be accounted for.
- Individual performance is independent of teammates and opposing players.
- There is a small inconsistency in the model. Drawing occurs whenever two teams have a performance that differs less than a chosen constant ϵ. This means that whenever team performance differs by $< \epsilon$ teams have the same rank (they draw), and whenever the difference $> \epsilon$ the better performing team has a lower (better) rank. We can see that for 3 teams, we can get the following inconsistency:

$$t_i = t_j + \frac{2}{3}\epsilon = t_k + \frac{2}{3}\epsilon \tag{49}$$

  Then we would have that i draws j, j draws k which implies i draws k, but we also have $t_i = t_k + \frac{3}{2}\epsilon > t_k + \epsilon$, thus team i should have a lower rank than team k from a standalone perspective. It is unclear how the model deals with this inconsistency.

2.7 Coalition Assessment in Film-Making

The final model we discuss was developed to estimate individual qualities of players performing in coalitions, but not necessarily in a competitive environment (Timmer *et al.* , 2017). The model was applied in the world of film-making to identify qualities of professionals and predict the potential quality of future projects. We translated formulation in the paper to fit the language of this thesis. Throughout this thesis, we referred to events where coalitions of players perform as matches. Rather than using the term *match* (which implies a competitive environment), we use the term *project* (which implies collaboration, i.e. all coalition members work together).
The idea is that players perform with an average quality and a Gaussian error. We have that the quality of a player i during project m is:

$$Q_{i,m} \sim \mathcal{N}(\mu_i, \sigma^2_{i,m}) \tag{50}$$

The variance $\sigma^2_{i,m}$ is model-specific. The variance can contain a factor that can increases variance for non-recent projects and decrease variance for experienced players.

Within a project, the players perform in teams (coalitions), we denote the set with the indices of players participating in project m by $C(m)$. The value of the team in project m, is denoted by V_m. It is assumed that player qualities can be aggregated by addition to yield the coalition value:

$$V_m = \sum_{i \in C(m)} Q_{i,m} \sim \mathcal{N}(\sum_{i \in C(m)} \mu_i, \sum_{i \in C(m)} \sigma^2_{i,m}) \tag{51}$$

The idea is that we have historical observations of player performance $Q^O_{i,m}$ for a set of historical projects. If only the value of a film is observed, V^O_m the authors suggest we can take:

$$Q^O_{i,m} = \beta_{i,m} V^O_m \tag{52}$$

Here $\beta_{i,m}$ denotes the importance of player i in project m according to some criterium. In the article, β_i is chosen to be the profit share of player (film-maker) i.
The general definition of a linear unbiased estimator $\hat{\mu}_i$ for the mean of the player quality μ_i is:

$$\hat{\mu}_i = \sum_{\{m | i \in C(m)\}} d_{i,m} Q^O_{i,m} \tag{53}$$

$$\text{Var}(\hat{\mu}_i) = \sum_{\{m | i \in C(m)\}} d^2_{i,m} \sigma^2_{i,m} \tag{54}$$

$$\sum_{\{m | i \in C(m)\}} d_{i,m} = 1 \tag{55}$$

The idea is that the weights d can be chosen, under the condition in Equation (55), such that the variance of our estimator is minimal, ensuring that $\hat{\mu}_i$ is the Best Linear Unbiased Estimator (BLUE). The optimal value of d depends on the exact definition of $\sigma^2_{i,m}$; the authors find explicit equations for specific choices of $\sigma^2_{i,m}$.
The model can also be applied in a setting where players have different weights in a coalition. The value of a team with weighted contributions is defined to be V^δ_m:

$$V^\delta_m = \sum_{i \in C(m)} \delta_{i,m} Q_{i,m} \sim \mathcal{N}(\sum_{i \in C(m)} \delta_{i,m} \mu_i, \sum_{i \in C(m)} \delta^2_{i,m} \sigma^2_{i,m}) \tag{56}$$

$$\sum_{i \in C(m)} \delta_{i,m} = 1 \tag{57}$$

This way the value of a project is not simply additive, but the importance of players is accounted for.

3 Mathematical model

This section presents the mathematical model we constructed to estimate player qualities by using multi-player team game data.
Players have multiple qualities, which are modeled as latent variables. We are interested in estimating these qualities, but they cannot be observed or measured directly. What we do observe are outcomes that follow from matches. Matches are interactions between coalitions of players. We refer to a coalition of players as a team. We call an observable quantity, related to a match, a Key Performance Indicator (KPI). KPI outcomes are a direct result of the performance of the participating players. The performances of players are non-deterministic but are closely related to the player quality we seek to estimate.
Our model contains four types of objects types; players, qualities, matches and KPI's. We will use indices to represent affiliation between variables and objects; we use i for players, j for qualities, m for matches and k for KPI's. We define N_P,N_Q, N_M and N_K as the number of players, qualities per player, matches and KPI's per match respectively. Qualities are related to players; therefore a specific quality of a player is referred to as player-quality. KPI's are related to matches; therefore a specific KPI of a match is referred to as match-KPI.
Within our model we make the following assumptions:

- a.1 The quality of a player is a normal random variable.
- a.2 The mean of the quality of a player is a normal random variable.
- a.3 The performance of a player is a realization of the random variable representing the player quality.
- a.4 Player performances are independent.
  - a.4.1 Performances over qualities of the same player are independent.
  - a.4.2 Performances over qualities of different players are independent.
  - a.4.3 Performances over a single quality of a player within a match for different KPI's are independent.
- a.5 The outcome space of KPI's is ordered and discrete (ordinal).
- a.6 The outcome distribution of a match-KPI follows directly from an aggregation of the performance, role, intention and participation, of players.
- a.7 Player performances can be aggregated by weighted summation. The result is a representation of the performance of the complete coalition.
  - a.7.1 Such weights are defined for all combinations of player, quality, match, and KPI.
- a.8 Players that have the intention to increase (decrease) a KPI have a positive (negative) weight factor.
- a.9 We have a method (designed using domain knowledge) that deterministically determines the influence on a KPI given a player's role, intention, and participation data.

The quality j of player i is represented by $Q_{(i,j)}$ and with assumptions a.1 and a.2 we have that;

$$Q_{(i,j)} \sim \mathcal{N}\left(M_{(i,j)}, \sigma^2_{(i,j)}\right) \tag{58}$$

$$M_{(i,j)} \sim \mathcal{N}\left(\mu_{M_{(i,j)}}, \sigma^2_{M_{(i,j)}}\right) \tag{59}$$

$$\sigma^2_{(i,j)} \in \mathbb{R}^+,\ \mu_{M_{(i,j)}} \in \mathbb{R},\ \sigma^2_{M_{(i,j)}} \in \mathbb{R}^+ \tag{60}$$

Here we have that $\mathcal{N}(a,b), a \in \mathbb{R}, b \in R^+$ is the Gaussian distribution with mean a and variance b.
We represent all qualities of player i in the vector $Q_{(i)}$, and all the qualities of all the players in the

vector Q. The distribution of these vectors is:

$$M_{(i)} \in \mathbb{R}^{N_Q \times 1}, \Sigma_{q_i} \in \mathbb{R}^{N_Q \times N_Q} \tag{61}$$

$$Q_{(i)} = \begin{bmatrix} Q_{(i,1)} \\ Q_{(i,2)} \\ ... \\ Q_{(i,N_Q)} \end{bmatrix} \sim \mathcal{N}(M_{(i)}, \Sigma_{q_i}) \tag{62}$$

$$M_q \in \mathbb{R}^{(N_Q N_P) \times 1},\ \Sigma_q \in \mathbb{R}^{(N_Q N_P) \times (N_Q N_P)} \tag{63}$$

$$Q = \begin{bmatrix} Q_{(1)} \\ Q_{(2)} \\ ... \\ Q_{(N_P)} \end{bmatrix} \sim \mathcal{N}(M_q, \Sigma_q) \tag{64}$$

We use that $\mathcal{N}(A, B)$ is the multivariate Gaussian distribution with mean A and covariance matrix B. Here we require that $A \in R^S, B \in R^{S \times S}$, and B is symmetric and positive semi-definite.
The vectors $M_{(i)}$ and M_q are random variables representing the mean of the qualities of player i and all the players, respectively. The matrices Σ_{q_i} and Σ_q are covariance matrices of the qualities of player i and the qualities of all players, respectively.
From Assumption a.4.1 we have that Σ_{q_i} is a diagonal matrix, as player qualities are uncorrelated. From Assumption a.4.2 we have that Σ_q is also a diagonal matrix. This is because independence implies the following relationship:

$$\text{Cov}(Q_{(a,b)}, Q_{(c,d)}) = 0 \text{ if } (a \neq c) \vee (b \neq d) \tag{65}$$

For the mean of the player qualities we find the following distribution:

$$M_q \sim \mathcal{N}(\mu_{M_q}, \Sigma_{M_q}) \tag{66}$$

$$\mu_{M_q} \in \mathbb{R}^{N_P N_Q \times 1},\ \Sigma_{M_q} \in \mathbb{R}^{N_P N_Q \times N_P N_Q} \tag{67}$$

We define $D_{k,m}$ as the random variable representing KPI k of match m. The realizations of the random variable $D_{k,m}$ is represented by $d_{k,m}$. All the KPI's of match m are represented by:

$$D_m = \begin{bmatrix} D_{1,m} \\ D_{2,m} \\ ... \\ D_{N_K,m} \end{bmatrix}, D = \begin{bmatrix} D_1 \\ D_2 \\ ... \\ D_{N_M} \end{bmatrix} \tag{68}$$

$$d_{i,m} \in \mathbb{D}_i \tag{69}$$

$$d_m \in \mathbb{D} = \mathbb{D}_1 \times \mathbb{D}_2 \times \ldots \mathbb{D}_{N_K}, m \in \{1, ..., N_M\} \tag{70}$$

$$\mathbb{D}_i \text{ is an ordinal set } \forall_i \tag{71}$$

The idea is that $D_{k,m}$ is a random variable with the outcome space $\mathbb{D}_k$.
In general, KPI outcomes can be both continuous and discrete, with very context specific probability distributions. As stated in Assumption a.5, we take KPI's to be ordinal. Each KPI type can have a different outcome space, therefore as stated in Equation (69) we have different outcome spaces for different KPI types. The complete probability model is discussed extensively in Section 4. Some examples of KPI's and non-KPI's are listed in Table 1.
During a match several players perform, but we only observe KPI's that are related to an aggregation of the performance of players (Assumption a.6). The idea is that the aggregation we perform is a representative way to measure team performance, and thus influences the outcome of the KPI. From Assumption a.7 we have that the performance of different players can be aggregated by using an importance weighted linear addition. We call such linear importance weights δ. Specific scaling factors are denoted by $\delta_{(i,j,k,m)}$ and depend on the player i, the quality j, the observed KPI k and the match m. We define $\delta_{(k,m)}$ as the weights associated with match m and KPI k, $\delta_{(m)}$ as the weights corresponding to match m and δ as the matrix with weights for all match-KPI's:

$$\delta_{(k,m)} = \begin{bmatrix} \delta_{(1,1,k,m)} & \delta_{(2,1,k,m)} & \cdots & \delta_{(N_P,1,k,m)} & \delta_{(1,2,k,m)} & \cdots & \delta_{(N_P,N_Q,k,m)} \end{bmatrix} \tag{72}$$

$$\delta_{(m)} = \begin{bmatrix} \delta_{(1,m)} \\ \delta_{(2,m)} \\ \cdots \\ \delta_{(N_K,m)} \end{bmatrix} = \begin{bmatrix} \delta_{(1,1,1,m)} & \delta_{(2,1,1,m)} & \cdots & \delta_{(N_P,1,1,m)} & \delta_{(1,2,1,m)} & \cdots & \delta_{(N_P,N_Q,1,m)} \\ \delta_{(1,1,2,m)} & \delta_{(2,1,2,m)} & \cdots & \delta_{(N_P,1,2,m)} & \delta_{(1,2,2,m)} & \cdots & \delta_{(N_P,N_Q,2,m)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \delta_{(1,1,N_K,m)} & \delta_{(2,1,N_K,m)} & \cdots & \delta_{(N_P,1,N_K,m)} & \delta_{(1,2,N_K,m)} & \cdots & \delta_{(N_P,N_Q,N_K,m)} \end{bmatrix} \tag{73}$$

$$\delta = \begin{bmatrix} \delta_{(1,1)} \\ \delta_{(2,1)} \\ \cdots \\ \delta_{(N_K,1)} \\ \delta_{(1,2)} \\ \cdots \\ \delta_{(N_K,N_M)} \end{bmatrix} = \begin{bmatrix} \delta_{(1,1,1,1)} & \delta_{(2,1,1,1)} & \cdots & \delta_{(N_P,1,1,1)} & \delta_{(1,2,1,1)} & \cdots & \delta_{(N_P,N_Q,1,1)} \\ \delta_{(1,1,2,1)} & \delta_{(2,1,2,1)} & \cdots & \delta_{(N_P,1,2,1)} & \delta_{(1,2,2,1)} & \cdots & \delta_{(N_P,N_Q,2,1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \delta_{(1,1,N_K,1)} & \delta_{(2,1,N_K,1)} & \cdots & \delta_{(N_P,1,N_K,1)} & \delta_{(1,2,N_K,1)} & \cdots & \delta_{(N_P,N_Q,N_K,1)} \\ \delta_{(1,1,1,2)} & \delta_{(2,1,1,2)} & \cdots & \delta_{(N_P,1,1,2)} & \delta_{(1,2,1,2)} & \cdots & \delta_{(N_P,N_Q,1,2)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \delta_{(1,1,N_K,N_M)} & \delta_{(2,1,N_K,N_M)} & \cdots & \delta_{(N_P,1,N_K,N_M)} & \delta_{(1,2,N_K,N_M)} & \cdots & \delta_{(N_P,N_Q,N_K,N_M)} \end{bmatrix} \tag{74}$$

$$\delta_{(i,j,k,m)} \in \mathbb{R},\ \delta_{(k,m)} \in \mathbb{R}^{1\times(N_P N_Q)},\ \delta_{(m)} \in \mathbb{R}^{N_K\times(N_P N_Q)},\ \delta \in \mathbb{R}^{(N_K N_M)\times(N_P N_Q)} \tag{75}$$

The value of $\delta_{(\cdot)}$ follows directly from the player's participation data: playing position, intention and playing time. As stated in Assumption a.9, we have a method to determine the importance weights from historical participation data. Such a method must be devised using domain knowledge, incorporating the fact that $\delta_{(i,j,k,m)}$ should be the importance weight of quality j of player i on KPI k in match m. In Section 6.1 we show the method we used for our application in football.
Note that $\delta_{(i,j,k,m)}$ is defined for every combination of i, j, k, and m, also in the case that a player does not even participate in a match. In such cases the player does not have any influence on the KPI's in match m, we define:

$$\text{Player } i \text{ does not participate in match } m \implies \delta_{(i,j,k,m)} = 0\ \forall_{j,k} \tag{76}$$

In most cases, only a fraction of all players participates in a match. In football, only 22-28 players are involved in a match, while our model considers a large population of players. Effectively this means that δ is a large sparse matrix. There might be environments where during each match a large proportion of players participates, consequently, the matrix δ will not be sparse. We can say that if during a match, on average N_Z players perform, each row of δ will contain (at most) $N_Z \cdot N_Q$ non-zero values. More δ's can be zero; as participating players that do not influence a KPI with a certain quality will receive $\delta_{(\cdot)} = 0$. It follows that the approximate fraction of non-zero elements in δ is $\frac{N_Z}{N_P}$.
Within a match players have certain intentions; they are constantly trying to achieve something. We model the intentions of players very straightforward, a player either wants to increase a KPI or decrease a KPI. This also follows from Assumption a.8. Whenever we have two teams, we have that players of opposing teams, with opposing intentions, will have weight factors $\delta_{(\cdot)}$ with opposite signs. Whenever aggregating, we are effectively calculating the strength difference between the players that want to increase and decrease a KPI. For this reason, we refer to the aggregation as the strength difference, dS. For KPI k in match m we define the strength difference as $dS_{k,m}$. We have that:

$$dS_{k,m} = \sum_{i,j} \delta_{(i,j,k,m)} Q_{(i,j)} = \delta_{(k,m)} Q \tag{77}$$

$$dS_{k,m} \in \mathbb{R} \tag{78}$$

As $dS_{k,m}$ is a weighted sum of Gaussian random variables, it is a Gaussian random variable itself. We have:

$$dS_{k,m} \sim \mathcal{N}(\delta_{(k,m)} M_q, \delta_{(k,m)} \Sigma_q \delta_{(k,m)}^T) \tag{79}$$

$$= \mathcal{N}(\delta_{(k,m)} \mu_{M_q}, \delta_{(k,m)} \Sigma_q \delta_{(k,m)}^T + \delta_{(k,m)} \Sigma_{M_q} \delta_{(k,m)}^T) \tag{80}$$

The randomness of $dS_{k,m}$ is caused by inconsistency in the performance and uncertainty in the mean quality of players. The realization of $dS_{k,m}$ is referred to as $dS_{k,m}^O$. We use the superscript "O", because $dS_{k,m}^O$ is an observation of $dS_{k,m}$.
From Assumption a.6 we assume that the probability distribution of such a KPI is a function of the

strength difference. We define a function $\xi_k : (\mathbb{D}_k \times \mathbb{R}) \implies [0,1]$ for each KPI k, that models the relationship between the strength difference and the outcome probabilities for KPI k:

$$P(D_{k,m} = d) = \xi_k(d, dS_{k,m}) \tag{81}$$

The idea of the function ξ is that it captures the relationship between the strength difference during a match, and the probability distribution of the KPI outcome. In Section 4 we will elaborate this relationship, which is essential to estimate past performance of players from historical data.
The average performance of players is a good indication of future player performance, but players will always overperform or underperform due to random effects. We call such deviations from expected performance *player quality inconsistencies*, these are captures by the matrix Σ_q. From Assumption a.4 we have that performance inconsistencies of players are uncorrelated. We conclude that inconsistencies in the strength differences of different match-KPI's, even KPI's in the same match, are uncorrelated. Even the performance inconsistency of a single player-quality for a different match-KPI is independent. For this assumption to be reasonable we must consider KPI's within a match, that are independent. We define the vector with all strength differences, dS, by:

$$dS \sim \mathcal{N}(\delta M_q, \delta \Sigma_q \delta^T \circ I) \tag{82}$$

In Equation (82) we use $\delta \Sigma_q \delta^T \circ I$, where $\circ$ is the element-wise product. By applying this operation we ensure that Σ_{dS} is a diagonal matrix. As required, the strength for match-KPI's are uncorrelated: $\text{cov}(dS_{a,b}, dS_{c,d}) = 0$, $\forall_{a \neq c \vee b \neq d}$. The fact that all strength differences are independent follows from all assumptions in a.4. Assumption a.4.3 is essential to ensure match-KPI's within a certain match are independent.
The main goal of our model is to calculate estimators for the probability distribution parameters of the individual player qualities; μ_{M_q}, Σ_{M_q} and Σ_q. These player qualities cannot be observed, and therefore are latent variables that we create to model player strengths. What we do observe are KPI realizations. The unknowns in our model are μ_{M_q}, Σ_{M_q}, Σ_q, δ and $\xi_i(\cdot)$. The values for $\xi_i(\cdot)$ and δ will be chosen in sections 4.1 and 6.1 respectively.

Table 1: Examples of KPI's; the main requirement is that is must be a measurable outcome of a collaborating team. Some examples have a continuous outcome space, to be applied in our model we would need to discretize the variable to yield an ordinal outcome space.

Context	KPI	Not KPI
Football	goals scored/conceded goals conceded shots on target	ability to attack ability to defend stamina
Tennis	first serve percentage points scored serving/not serving	player length public attendance
Golf	score relative to par	player gear
Start-up	sales	company culture
Film-making	IMDB score box office sales	movie review text actors

Remark 2. *In Section 2 we described multiple models, we list the main characteristics of our model to highlight the differences:*

- *Any amount of players can participate in a match*
- *Matches have multiple observable outcomes*
- *Match outcomes are ordinal*
- *Players have multiple qualities*
- *The mean of a player's quality is modeled as a random variable*

Remark 3. *Our model shares the assumption that player qualities are additive with (Timmer* et al. *, 2017), Section 2.7, and (Herbrich* et al. *, 2007), Section 2.6. We can say that player qualities are part of a linear space where multiple qualities can be aggregated by addition. This holds for our model and both these models. The main difference is the space of our observations. In (Timmer* et al. *, 2017) the*

outcome space is the same as the quality space; after adding all qualities no further transformations are applied. In (Herbrich et al. *, 2007) our observations are a ranking of the team qualities; which is a non-linear transformation of elements of the space. The idea in our research is that we assume additive player qualities, and a non-linear relationship with the space of our observations. Up to this point, this relationship has been defined in Equation* (81) *by an unspecified mapping* ξ.

4 Probability model

In this section, we will construct the probability model that describes the relationship between player qualities and outcomes of matches.
We will discuss our choices that lead to an expression for the conditional outcome probability $P_\theta(D_m = d_m)$. We contain all relevant information regarding a match in the parameter θ. Essentially what we will be modeling is the function ξ from Equation (81), as this is the relationship between the player qualities (aggregated in dS) and realizations d of D.
Recall from Assumption a.5 that our KPI outcomes, D, are ordinal. A variable is ordinal if the outcome space is countable and ordered. An example is the Likert scale (Likert, 1932), an educational grading system or the outcome of a football match.
Ideas from the field of psychometrics were found to be applicable to our research. This field deals with measurements of latent characteristics based on comparison data. Ordinal outcomes are very common in this field (Casalicchio, 2013). Whenever pairwise comparisons are performed by humans, there are underlying stimuli that guide the decision-making process. The strength of these stimuli can be modeled as a latent variable, just like we have done with the player ratings in our models.
Throughout this section, we will develop methodologies and formula's that hold for a single KPI within one match, and can easily be extended to multiple KPI's for a large set of matches. Therefore, in this section, we will refer to KPI's only with a single index (defining the matches), while in the rest of the thesis we apply a double index (defining the match and specific KPI type).
In Section 4.1 we will look at the general method behind parametrizing the outcome probability distribution conditioned on player strength. In this subsection, we will introduce three abstract concepts and discuss them in the next subsections: utility function 4.2.1, benchmark parameters 4.2.2 and link function 4.2.4. In Section 4.3 we will discuss how we observe the strength difference within a match, and we will continue to explain how we can use bookmaker odds as an input to our model in Subsection 4.4.

4.1 Relationship between KPI outcomes and strength difference

We want to find a model for the probability distribution of KPI's given the player-data in a match (player-roles, participation and intentions). We already have the first step; we defined the strength difference dS as the aggregation of player strengths on a match level in Equation (77). Even though for a match multiple KPI's can be observed, as stated, in this section we will use the index m to specify a specific KPI for match m (rather than the double index convention used in Section 3).
Our KPI's are random variables, D_m, with realizations d_m that are elements of an ordinal set $\mathbb{D}_{(\cdot)}$, as defined in Equation (69). The index of $\mathbb{D}$ is unspecified, we use to the outcome space of the KPI under consideration. We say $\mathbb{D}_{(\cdot)} = \{1, 2, ..., K\}$, thus it is a set of K ordered outcomes. From Equation (81) we have that the probability distribution of D_m is regulated by the strength difference within match-KPI m, dS_m, as stated in Equation (81). Recall from Equation (77) that we have

$$dS_m = \delta_m Q \tag{83}$$
$$\mu_{dS_m} = E[dS_m] \tag{84}$$
$$\sigma^2_{dS_m} = E[dS_m^2] - E[dS_m]^2 \tag{85}$$

In the next sections, we construct a direct mapping from the realization of the strength difference, dS_m^O, to the realized outcome d_m. We used methodologies from psychometrics, with some specific choices that lead to the following relationship:

$$\{dS_m^O \in [\beta_{k-1}, \beta_k)\} \iff \{d_m = k\} \tag{86}$$
$$P(d_m = k) = P(dS_m^O \in [\beta_{k-1}, \beta_k)) \tag{87}$$

This means that we have a relationship between regions of the realized strength difference, $[\beta_{k-1}, \beta_k)$, and KPI outcomes. Whenever the strength difference realization falls in such an interval, the outcome d_m of the match is the corresponding ordinal element of $\mathbb{D}$.
We use the following notation for the outcome probability distribution, conditioned on the distribution of the strength difference:

$$P(D_m = d_m | dS_m \sim N(\mu_{dS_m}, \sigma^2_{dS_m})) \tag{88}$$

The idea behind this notation is that in matches can be very different (mostly due to the fact that different players participating), and outcome probability distributions are a function of the strength difference, dS_m, distribution.

4.2 IRT model

The modeling approach we apply comes from the field of Item Response Theory (IRT). This theory finds a probabilistic relationship between performances of individuals on specific tests and a measure of their general performance in other tests. The goal of an IRT model is to describe the relationship between latent traits (unobservable inputs) and (observable) experiment outcomes. The main idea is that the outcome of an experiment depends on underlying traits, that can be inferred by using the experiment outcome. The formulation of our model is abstract and contains multiple unknown elements. In the subsequent subsections, we will elaborate how we constructed or chose these unknown elements. The approach is similar to (McCullagh, 1980), and has been developed further by (Fahrmeir & Tutz, 1994) and (Casalicchio, 2013).
We use a utility function $u : \mathbb{R} \to \mathbb{R}$, a continuous monotonically increasing function, and a link function $F : \mathbb{R} \to [0, 1]$, a cumulative distribution function. The utility function reflects the importance of a certain strength difference. The link function describes the relationship between the outcome and an affine transformation of the utility strength difference. Furthermore, we use the following:

$$E[dS_m] = \mu_{dS_m} \in \mathbb{R} \tag{89}$$
$$\text{Var}(dS_m) = \sigma^2_{dS_m} \in \mathbb{R}^+ \tag{90}$$
$$\beta_k \in \mathbb{R} \text{ for } k \in \{0, ..., K\} \tag{91}$$
$$\{i \geq j\} \implies \{\beta_i \geq \beta_j\} \tag{92}$$
$$\beta_0 = -\infty \wedge \beta_K = \infty \tag{93}$$
$$u(0) = 0 \tag{94}$$

We get the following equation:

$$P(D_m \leq k | dS_m \sim \mathcal{N}(\mu_{dS_m}, \sigma^2_{dS_m})) = F(\beta_k - u(\mu_{dS_m}), \sigma_{dS_m}). \tag{95}$$

By defining the above we see that the ordinal response D has a probability distribution that depends on the distribution of strength discrepancy $dS_m \sim \mathcal{N}(\mu_{dS_m}, \sigma_{dS_m})$. The strength discrepancy utility is equal to $u(\mu_{dS_m})$.
If we would assume that for all samples the second moment is equal, we can let it be absorbed by the function F. Under this assumption, equation (95) is a specific case of the generalized Fechner Problem, where $F(\cdot)$ is Fechnerian discrimination index and $u(\cdot)$ is a utility function (Falmagne, 1971). Fechner's research was pioneering in finding mathematical measures that represent the psychometric principles behind how humans distinguish objects. Their idea was to find mappings between object features and preferences. We can extend the usage of these ideas to the context of player comparisons in any competitive environment.
We choose to add randomness to the psychometric model, because the performance of players in a competitive environment is affected by a lot of random effects. Also it makes sense that results are more (less) predictable, if we have a higher (lower) certainty regarding the strength difference.
It is clear that the strength difference has a match specific probability distribution.

Example 1. *When considering the KPI home goals scored in football, it is clear that the home team wants to increase this KPI. From Assumption a.8 it follows that the home players will have $\delta_{(\cdot)} > 0$ for this KPI. The parameters of the strength difference dS will be different, depending on the quality of the players of the two teams. We can say that $u(\mu_{dS_m}) \approx 0$ (no strength difference) in matches where teams are evenly matched, while in a match where the home team is stronger we would have $u(\mu_{dS_m}) > 0$. From Equation (94) we have $u(\mu_{dS_m}) = 0 \implies \mu_{dS_m} = 0$, combining this with the fact that u is monotonously increasing we have $u(\mu_{dS_m}) > 0 \implies \mu_{dS_m} > 0$.*

We can speak of *the average strength difference* and *the average inconsistency* for matches, which are the population means. We define $\bar{\mu}_{dS}$ as the population mean of μ_{dS}, and $\bar{\sigma}^2_{dS}$ as the population mean of σ^2_{dS}. These numbers can be interpreted as the mean and variance of *an average match.* The utility of the average strength difference, $u(\bar{\mu}_{dS})$, should be 0 as on average there is no strength difference between teams. We get $\bar{\mu}_{dS} = 0$ from Equation (94) and we choose $\bar{\sigma}^2_{dS} = 1$, without loss of generality. Our choices of the population mean and population variance are free because dS is a latent variable that is translation and scale invariant. Also, any scaling and translations of dS can also be absorbed by alternative, methodologically equivalent, choices for β, the functions u and the function F.
We now proceed to analyze Equation (95) under the assumption that $\sigma^2_{dS_m} = \bar{\sigma}^2_{dS} = 1$. We will use

the following notation:

$$\bar{F}(x) = F(x, \bar{\sigma}^2_{dS}). \tag{96}$$

From this we get the following equation:

$$\beta_i = \bar{F}^{-1}\left(\sum_{j=1}^{i} P\left(D_m = j | dS_m \sim \mathcal{N}\left(\bar{\mu}_{dS}, \bar{\sigma}^2_{dS}\right)\right)\right), \text{ for } i = 0, 1, 2, ..., K. \tag{97}$$

Note that we have $\beta_0 = -\infty$ and $\beta_K = \infty$. From (95) it follows directly that:

$$P(D_m = k | dS_m \sim \mathcal{N}(s, \bar{\sigma}^2_{dS})) = \bar{F}(\beta_k - u(s)) - \bar{F}(\beta_{k-1} - u(s)). \tag{98}$$

Recall that F and $\bar{F}$ are cumulative density functions. In the case these functions are differentiable they have well defined densities, $f(x, r^2) = \frac{\mathrm{d}F(x,r^2)}{\mathrm{d}x}$ and $\bar{f}(x) = \frac{\mathrm{d}\bar{F}(x)}{\mathrm{d}x}$, the corresponding probability mass functions. We note that by the fundamental theorem of calculus the following relationships hold:

$$P(D_m = k | dS_m \sim \mathcal{N}(s, r^2)) = \int_{\beta_{k-1}+u(s)}^{\beta_k+u(s)} f(x, r^2)dx \tag{99}$$

$$= \int_{\beta_{k-1}}^{\beta_k} f(x + u(s), r^2)dx \tag{100}$$

$$P(D_m = k | dS_m \sim \mathcal{N}(s, \bar{\sigma}^2_{dS})) = \int_{\beta_{k-1}+u(s)}^{\beta_k+u(s)} \bar{f}(x)dx \tag{101}$$

$$= \int_{\beta_{k-1}}^{\beta_k} \bar{f}(x + u(s))dx \tag{102}$$

The function $u(s)$ represents the impact of a strength difference of s. Our choice for this function is elaborated in 4.2.1. In the Subsection 4.2.2, we will discuss an estimation procedure for β_i. In Section 4.2.4, we will assume that F is a normal cumulative distribution and show the implications this has. In Subsection A.1.2 of the appendix, we show the quality of the approximations when applied to estimate a Poisson distributed random variable.

4.2.1 Strength difference utility function

In the original formulation in the field of psychometrics the functional u represented the utility of a certain stimulus in a pairwise comparison. In our model, the inputs are individual player qualities and their influences on the comparison outcome. u is a mapping from these individual player qualities to their utility when it comes down to influencing a certain KPI outcome. We assume that this relationship is linear, we choose u to be the identity transformation and define dS_m according to Equation (77) as a combination of individual player qualities:

$$dS_m = \delta_m Q \tag{103}$$

$$\mu_{dS_m} = \delta_m \mu_{M_q} \tag{104}$$

$$\sigma^2_{dS_m} = \delta_m(\Sigma_{M_q} + \Sigma_q)\delta_m^T \tag{105}$$

$$u(\mu_{dS_m}) = \mu_{dS_m} \tag{106}$$

We can argue that the calculation of dS_m, by weighing the factors of Q with carefully chosen $\delta_{(\cdot)}$'s is already a linear transformation from player qualities to their utility. Therefore dS_m had already information about the utility of the quality of participating players, which is an argument for choosing u to be the unitary transform.
Note that $u(\cdot)$ this is not an increasing function with respect to the player qualities, Q, for elements of $Q_{(\cdot)}$ where the corresponding $\delta_{(\cdot)} < 0$ the function is strictly decreasing. It is trivial that $u(\cdot)$ is, as required, a monotone increasing function of dS_m. This is important because it ensures that a higher (lower) strength difference always gives a higher (lower) utility.

4.2.2 Benchmark parameter estimation

In this section we will determine estimators for the benchmarks β_k, using the complete dataset and Equation (95). An important assumption we make is that β_i's are constant wrt $\sigma^2_{dS_m}$ and μ_{dS_m}.
The benchmarks are KPI dependent; so in this subsection m refers to a certain KPI within match m. In total, we have N_K KPI types, and we apply the equations in this section for all KPI types separately.
We start off by adjusting equation (95) such that it holds (approximately) for the complete dataset, using the utility function chosen in Equation (106):

$$F(\beta_k, \bar{\sigma}^2_{dS_m}) = \bar{F}(\beta_k) = P(D_m \leq k | dS_m \sim \mathcal{N}(0, \bar{\sigma}^2_{dS})) \tag{107}$$

$$= P(D_m \leq k | dS_m \sim \mathcal{N}(\bar{\mu}_{dS}, \bar{\sigma}^2_{dS})) \tag{108}$$

$$\approx P(D_m \leq k) \tag{109}$$

This approximation is by definition not exact. In Equation (108) the outcome probability is conditioned on the average strength difference distribution, thus *averagely matched* teams, and in equation (109) we do not condition (effectively we consider a random match). The approximation follows from the assumption that match outcome probabilities for an average match, with an average strength difference ($\bar{\mu}_{dS} = 0$) and average variance ($\bar{\sigma}^2_{dS}$), are equal to the outcome probabilities of a random match. Therefore we can equate these outcome probabilities, and we can estimate outcome probabilities for equally matched teams by looking at the population average outcome occurrence frequency. Under the approximation in Equation (109) we yield:

$$\beta_k = \bar{F}^{-1} \left(\frac{\sum_{m=1}^{N_M} \mathbb{1}(d_m \leq k)}{N_M} \right). \tag{110}$$

Here we use the indicator function with a statement X as an argument:

$$\mathbb{1}(X) = \begin{cases} 1 & \text{if } X \text{ is True} \\ 0 & \text{if } X \text{ is False} \end{cases} \tag{111}$$

4.2.3 Improved benchmark parameter estimation

The benchmark parameter estimation in Subsection 4.2.2 was done using the dataset under the approximation in Equation (109). Once we have model results, we can relate strength difference estimates to outcome data. This enables us to find improved estimates of β_i; for example, we can make β_i estimates depend on μ_{dS_m}. The idea is that we yield more accurate estimates of β, for different average strength differences μ_{dS}. As we have the implied strength differences of our model, we can improve the approximation in (109) by taking:

$$P(D_m \leq k | dS_m \sim \mathcal{N}(0, \bar{\sigma}^2_{dS})) \approx P\big(D_m \leq k \big| \ |\mu_{dS}| < \epsilon\big) \tag{112}$$

Here epsilon is a small value, that ensures that we estimate β by only looking at matches that according to our model are evenly matched. This can be seen as a clustering method, that we can use to calculate β_i for all possible strength differences by defining:

$$\beta_k(s) = \bar{F}^{-1} \left(\frac{\sum_{m=1}^{N_M} \mathbb{1}(|\mu_{dS_m} - s| < \epsilon) \mathbb{1}(d_m \leq k)}{\sum_{m=1}^{N_M} \mathbb{1}(|\mu_{dS_m} - s| < \epsilon)} \right) \tag{113}$$

Here $\beta_k(s)$ is used for a match whenever $\mu_{dS_m} = s$.
Another method that can be applied is Kernel Density Filtering (KDF), yielding the following formula:

$$\beta_k(s) = \bar{F}^{-1} \left(\frac{\sum_{m=1}^{N_M} \mathbb{1}(d_m \leq k) \mathcal{N}(s; \mu_{dS_m}, \sigma^2_{dS_m})}{\sum_{m=1}^{N_M} \mathcal{N}(s; \mu_{dS_m}, \sigma^2_{dS_m})} \right) \tag{114}$$

This way samples are weighted accordingly with a Gaussian kernel. We choose the kernel to be the match dependent probability distribution of the strength difference; $\mathcal{N}(\mu_{dS_m}, \sigma^2_{dS_m})$.

4.2.4 Gaussian distribution as link function

In this section we apply the model from 4.1 and choose an appropriate $F(\cdot,\cdot)$. Such a function is often referred to as the link function; as it is the function that links the input variables to the output variables. We choose F to be the normal cumulative distribution, we get $F(a,b) = \Phi(\frac{a}{\sqrt{b}}) = P(x < \frac{a}{\sqrt{b}} | x \sim \mathcal{N}(0,1))$. Here we have that Φ is the standard normal cumulative distribution. From 4.2 we have that $\bar{\sigma}^2_{dS} = 1$, thus it follows that $\bar{F}(a) = F(a,1) = \Phi(a)$. We get:

$$P(D_m \leq k | dS_m \sim \mathcal{N}(s, r^2)) = F(\beta_k - u(s), r^2) \tag{115}$$

$$= \Phi\left(\frac{\beta_k - s}{r}\right) \tag{116}$$

The model has a lot of resemblances with the Thurstone-Mosteller model from Subsection 2.3.2, but where this model uses binary outcomes we have ordinal outcomes.
We now use equations (99), (100) and (106) yielding the following for $k \neq 1$ and $k \neq K$:

$$P(D_m = k | dS_m \sim \mathcal{N}(s, r^2)) = \Phi\left(\frac{\beta_k - s}{r}\right) - \Phi\left(\frac{\beta_{k-1} - s}{r}\right) \tag{117}$$

$$= \int_{\beta_{k-1}}^{\beta_k} \mathcal{N}(x; s, r^2) dx \tag{118}$$

$$= \int_{-\infty}^{\infty} \mathbb{1}_{[\beta_{k-1}, \beta_k]}(x) \mathcal{N}(x; s, r^2) dx \tag{119}$$

$$= \int_{-\infty}^{\infty} \mathbb{1}_{[-\beta_k, -\beta_{k-1}]}(-x) \mathcal{N}(x; s, r^2) dx \tag{120}$$

$$= |\beta_k - \beta_{k-1}| \int_{-\infty}^{\infty} \mathcal{U}[-\beta_k, -\beta_{k-1}](0 - x) \mathcal{N}(x; s, r^2) dx \tag{121}$$

$$= |\beta_k - \beta_{k-1}| \left(\mathcal{U}[-\beta_k, -\beta_{k-1}] * \mathcal{N}(s, r^2)\right)(0) \tag{122}$$

Here we use the indicator function and the continuous uniform probability density function:

$$\mathbb{1}_{[A,B]}(x) = \begin{cases} 1 & \text{if } x \geq A \wedge x \leq B \\ 0 & \text{if } x < A \vee x > B \end{cases} \tag{123}$$

$$\mathcal{U}[C,E](x) = \begin{cases} \frac{1}{|B-A|} & \text{if } x \geq C \wedge x \leq E \\ 0 & \text{if } x < C \vee x > E \end{cases} \tag{124}$$

$$= \frac{1}{|C-E|} \mathbb{1}_{[C,E]}(x) \tag{125}$$

Furthermore, to get from Equation (121) to (122) we use the definition of the convolution of two probability distributions:

$$h(z) = (f * g)(z) = \int_{-\infty}^{\infty} f(z - t) g(t) dt \tag{126}$$

We know from (Grinstead & Snell, 2009) that the convolution of two probability density functions corresponds to the pdf of a random variable that is distributed as the sum of the two underlying random variables. We now yield that:

$$P(D = k | dS \sim \mathcal{N}(s, r^2)) \sim (\beta_k - \beta_{k-1}) p_C(0) \tag{127}$$

$$C = A + B \tag{128}$$

$$A \sim U[-\beta_k, -\beta_{k-1}] \tag{129}$$

$$B \sim \mathcal{N}(s, r^2) \tag{130}$$

We would like to emphasize that this expression for $P(D_m = k | dS_m \sim \mathcal{N}(s, r^2))$ does not integrate to 1 as it is not a probability distribution, but a probability of an event happening conditioned on

the distribution dS_m. Therefore we do have that

$$\sum_k P(D_m = k|dS_m \sim \mathcal{N}(s, r^2)) = 1$$

We can now approximate $P(D_m = k|dS_m \sim \mathcal{N}(s, r^2))$ probability for $k = \{2, 3, ..., K-1\}$ by a normal distribution. We do this by a method called assumed density filtering, which in our case comes down to moment matching (Ranganathan, 2004), because our assumed density function is the normal distribution. This means that our normal approximation is optimal according to this methodology when we choose the corresponding 1st and 2nd moments:

$$E[C] = E[A] + E[B] = -\frac{\beta_k + \beta_{k-1}}{2} + s \tag{131}$$

$$\text{Var}(C) = \text{Var}(A) + \text{Var}(B) = \frac{(\beta_k - \beta_{k-1})^2}{12} + r^2 \tag{132}$$

We see that for $k = 1$ or $k = K$, the moments are not well defined ($\beta_0 = -\infty$ and $\beta_K = \infty$), therefore we will need a different calculation for these cases.
From Equations (131) and (132) we have seen that we can approximate $P(D_m = k|dS_m \sim \mathcal{N}(s, r^2))$, for $k = \{2, 3, ..., K-1\}$, by the following Gaussian distribution:

$$P(D_m = k|dS_m \sim \mathcal{N}(s, r^2)) = \Phi\left(\frac{\beta_k - s}{r}\right) - \Phi\left(\frac{\beta_{k-1} - s}{r}\right) \tag{133}$$

$$\approx (\beta_k - \beta_{k-1})\mathcal{N}\left(0; s - \frac{\beta_k - \beta_{k-1}}{2}, r^2 + \frac{(\beta_k - \beta_{k-1})^2}{12}\right) \tag{134}$$

$$= (\beta_k - \beta_{k-1})\mathcal{N}\left(-s; -\frac{\beta_k - \beta_{k-1}}{2}, r^2 + \frac{(\beta_k - \beta_{k-1})^2}{12}\right) \tag{135}$$

$$= (\beta_k - \beta_{k-1})\mathcal{N}\left(s; \frac{\beta_k - \beta_{k-1}}{2}, r^2 + \frac{(\beta_k - \beta_{k-1})^2}{12}\right) \tag{136}$$

$$\propto \mathcal{N}(s; \mu_{D_k}, r^2 + \sigma^2_{D_k}) \tag{137}$$

$$\mu_{D_k} = \frac{\beta_k - \beta_{k-1}}{2} \tag{138}$$

$$\sigma^2_{D_k} = \frac{(\beta_k + \beta_{k-1})^2}{12} \tag{139}$$

Now for the edge cases we have that:

$$P(D_m = k|dS_m \sim \mathcal{N}(s, r^2)) = \begin{cases} \Phi(\frac{\beta_1 - s}{r}) & \text{if } k = 1 \\ 1 - \Phi(\frac{\beta_{K-1} - s}{r}) & \text{if } k = K \end{cases} \tag{140}$$

4.3 Observed strength difference

In sections 3 and 4.2.1 we have defined strength difference for KPI k in match m as a linear weighted sum of the individual qualities of players, $dS_{k,m} = \delta_{k,m}Q$. Even though a team has a higher (average) quality, random factors can influence the performance and therefore the outcome of a match.
Whenever we observe the KPI outcome, we know that the $dS^O_{k,m}$ is in a certain range. We repeat Equation (86) that describes this relationship explicitly:

$$\{dS^O_{k,m} \in [\beta_{z-1}, \beta_z)\} \iff \{d_{k,m} = z\} \tag{141}$$

Essentially we never observe a specific value value of the strength difference, but we only know the prior distribution and the eventual range. We say that:

$$\left(dS^O_m \middle| D_m = k\right) = \left(dS^O_m \middle| dS^O_m \in [\beta_{k-1}, \beta_k)\right) \tag{142}$$

We see how this works in figure 1. In this example we have chosen $\mu_{dS} = 1$ and $\sigma^2_{dS} = 1$, and all filled regions under the normal distribution correspond to ordinal outcome bins. Whenever an outcome occurs, the observed strength difference for this match was somewhere in the bin, and the conditional distribution takes the form of a truncated Gaussian. To avoid dealing with truncated Gaussians we approximate such outcome with a normal distribution by moment matching. The equations for this are described in Subsection C.1. In figure 1 we show how this approximation looks in a specific

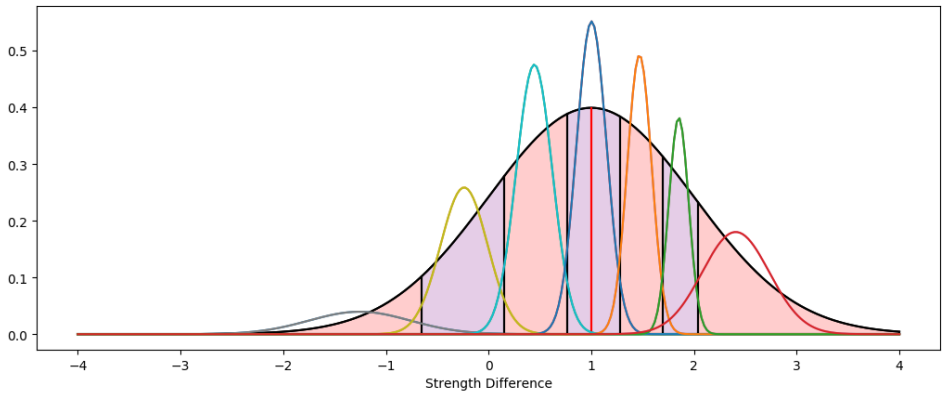

Figure 1: Standard class distribution

case. The figure displays Gaussian curves are an approximation of a truncated Gaussian areas. The Gaussian curves are normalized to have the same area as the area of the truncated Gaussian they approximate.
The following equations describe the mean and variance associated with a truncated Gaussian, and therefore our approximation:

$$p_{dS}(dS_m^O) \sim \mathcal{N}(dS_m^O; \mu_{dS_m}, \Sigma_{dS_m}) \tag{143}$$

$$p_{dS}\left(dS_m^O \middle| D_m = k\right) \sim \mathcal{N}(dS_m^O; \psi_k, \pi_k^2) \tag{144}$$

$$\psi_k = \mu_{dS} + \frac{\phi(a) - \phi(b)}{\Phi(b) - \Phi(a)}\sigma_{dS} \tag{145}$$

$$\pi_k^2 = \sigma_{dS}^2 \left[1 + \frac{a\phi(a) - b\phi(b)}{\Phi(b) - \Phi(a)} - \left(\frac{\phi(a) - \phi(b)}{\Phi(b) - \Phi(a)}\right)^2\right] \tag{146}$$

$$a = \frac{\beta_{k-1} - \mu_{dS}}{\sigma_{dS}} \tag{147}$$

$$b = \frac{\beta_k - \mu_{dS}}{\sigma_{dS}} \tag{148}$$

$$\phi(x) = \mathcal{N}(x; 0, 1) \tag{149}$$

$$\Phi(z) = P(X < z) \text{ where } X \sim \mathcal{N}(0, 1) \tag{150}$$

Almost all estimation techniques require observations to be certain, rather than a probability distribution. Therefore, for simplicity, throughout this report we will use a point estimate; $(dS_m^O | D_m = k) = \psi_k$. This assumption will not influence estimators of quality means, it will only affect our certainty of these means. In essence; we assume that our observations are point values of the strength difference, which corresponds to the average of the strength difference distribution associated with the KPI outcome. We discuss a method that could potentially be used to transfer the uncertainty of the observation to the parameter we estimate in Section C.3.

4.4 Market knowledge

In some cases a match between teams catches the interest of a lot of people. In general, disagreements about future outcomes are very common, and this leads to a large demand for money wagering. For individuals it provides an interesting way of fan engagement and companies can use it as investment opportunities. Fact is that there is a large market where people from the whole world bet on more than 1.000 events every day. Such bets are accessible to be wagered on by anyone with internet. The total annual gross win by companies in Sports Betting is 11.5 Billion in 2012 (Statista, 2012). Multiple sources indicate that this is only a fraction of the total sports betting market, as most action occurs illegally and therefore is unregistered.
The business model of companies in this market is to provide quotes (odds) that can be wagered on. Such odds indicate the multiple a customer receives whenever he wagers money on an event and this event happens. Just like in classical casino games (blackjack, roulette, etc.), the bookmakers aim to

provide a circumstance wherein the pay-off structure is such that their profit has a positive expected value. This means that in the long run they make profit consistently. The odds that bookmakers provide are the reciprocal of the implied probability. There are reliable methods to infer probabilities that bookmakers expect for an event from the odds they provide. The bookmakers have incentive to provide correct odds, otherwise there are players that can consistently take advantage of inefficiencies and make money in the long run. This means that the bookmaker will lose money; which is a scenario that they clearly want to avoid.

It must be noted that bookmaker odds are heavily influenced by (irrational) betting patterns of gamblers. This means that the odds in the market can always contain bias. Nevertheless; for events with high liquidity (a large amount of betting) we can use the "Efficient Market Hypothesis" to argue that the final implied probabilities, the equilibrium of all the participating parties, in general is a very good estimator of the real probabilities. The Efficient Market Hypothesis states that in financial economics all asset prices fully reflect all the information available. A sports bet could be seen as an investment opportunity; therefore wrong implied probabilities lead to mispricing and should be exploited by the rest of the market in the long term.

The bookmaker implied probabilities can be used to calculate bookmaker implied strength difference. An idea is to use predictions made by bookmakers as the measurements we train our model on. This way we use observations that could be biased, but contain much lower variance. Another advantage of this approach is that we do not need to make any mathematical model for $P_\theta(D_m = k)$, where θ contains all the relevant information for match m. This information is simply implied by the betting market. It is a result of all the people and companies in the world that have a certain opinion about the outcome of this match. We develop an estimation procedure using this methodology in Subsection 5.7.

5 Estimators for mean quality, quality variance and player inconsistency

In this section we will construct multiple estimators that may be used to find the parameters in our model. We are mostly interested in the mean player quality, M_q. This is parametrized by it's mean μ_{M_q} and uncertainty Σ_{M_q}. This player quality is the average player performance of a single player, while the inconsistency of the player performance is captured by the parameter Σ_q. We have the following probabilty model:

$$p_Q(q) = \mathcal{N}(q; M_q, \Sigma_q) \tag{151}$$
$$p_{M_q}(x) = \mathcal{N}(x; \mu_{M_q}, \Sigma_{M_q}) \tag{152}$$
$$p_{dS}(y|M_q = x) = \mathcal{N}(y; \delta x, \Sigma_{dS}) \tag{153}$$

We have that the variances of the strength differences for KPI outcomes, given M_q, are uncorrelated. We get:

$$\Sigma_{dS} = diag(\sigma^2_{dS_1}, \sigma^2_{dS_2}, ..., \sigma^2_{dS_{N_{MK}}}) = \begin{bmatrix} \sigma^2_{dS_1} & 0 & ... & 0 \\ 0 & \sigma^2_{dS_2} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \sigma^2_{dS_{N_{MK}}} \end{bmatrix} \tag{154}$$

$$\sigma^2_{dS_j} = \sum_{i=1}^{N_{PQ}} \delta^2_{j,i}(\Sigma_q)_{i,i} = \delta_{j,*}\Sigma_q\delta^T_{j,*} \tag{155}$$

$$\Sigma_{dS} = \delta\Sigma_q\delta^T \circ I \tag{156}$$

$$\Sigma_q = diag(\sigma^2_{q_1}, \sigma^2_{q_2}, ..., \sigma^2_{q_{N_{PQ}}}) = \begin{bmatrix} \sigma^2_{q_1} & 0 & ... & 0 \\ 0 & \sigma^2_{q_2} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \sigma^2_{q_{N_{PQ}}} \end{bmatrix} \tag{157}$$

Recall from Section 3, that N_M is the number of matches, N_K is the number of KPI's per match, N_Q is the number of qualities per player and N_P is the number of players in our dataset. For convenience we define $N_{KM} = N_K N_M$ and $N_{PQ} = N_P N_Q$. We are interested in finding estimates for μ_{M_q}, Σ_{M_q} and Σ_q, denoted by $\hat{\mu}_{M_q}$, $\hat{\Sigma}_{M_q}$ and $\hat{\Sigma}_q$ respectively. Estimators have a superscript that contains an abbreviation of the estimation method.
In the case that $\delta \in \mathbb{R}^{N_K N_M \times N_P N_Q} = \mathbb{R}^{N_{KM} \times N_{PQ}}$ has incomplete column rank this means that there exists collinearity between player performances. A statistical model cannot distinguish between collinear player performance without prior information or additional assumptions. Some methods require that δ has full row rank, implicitly requiring that $N_{KM} > N_{PQ}$.
Throughout this section we will find estimators that are optimal according to a certain criterion. The likelihood of an estimator is a very important property, defined in (1). Estimators that are optimal according to this criterion are called *maximum likelihood estimators* (MLE), defined in Equation (2) and (3). Another well known measure for the quality of an estimator is *the mean squared error* (MSE), defined as:

$$\text{MSE}\left(\hat{\theta}\right) = E\left[\left(\theta - \hat{\theta}\right)^2\right] \tag{158}$$

The estimator of θ that minimizes the MSE is referred to as the *minumum mean square error* (MMSE). The following properties and definitions are useful:

$$\text{MSE}\left(\hat{\theta}\right) = \text{Var}\left(\hat{\theta}\right) + \text{bias}\left(\hat{\theta}, \theta\right)^2 \tag{159}$$
$$\text{bias}\left(\hat{\theta}, \theta\right) = E\left[\hat{\theta}\right] - \theta = E\left[\hat{\theta} - \theta\right] \tag{160}$$
$$\text{MMSE}\left(\theta\right) = \underset{\hat{\theta}}{\text{argmin}}\, \text{MSE}\left(\hat{\theta}\right) \tag{161}$$
$$= E\left[\theta | D = d\right] \tag{162}$$

We prove Equation (159) in the Appendix B.1 and Equation (162) in the Appendix B.2. In Subsection 5.1 we calculate the likelihood function of our model, in an attempt to calculate maximum

likelihood estimators, using the formulas from Section 4.2.4. After this we will develop estimators that are unbiased in Subsections 5.2 and 5.3. Later, in sections 5.4 and 5.6 the estimates will be biased but with a lower MSE than the unbiased estimator. In the Subsection 5.5 and 5.6.1 we will discuss how the earlier discussed methods can be applied in batches. This allows us to yield intermediate results of our estimators. We will also go in-depth how we go about the estimation of player inconsistencies under the assumption that these are heteroskedastic in Appendix D. Finally; in Section 5.9 we summarize the results and explain which method we implemented.

5.1 Maximum likelihood estimation

As stated, a logical approach would be to find estimators for the parameters of our model that maximize the outcome likelihood. We get such MLE estimators by:

$$\mathcal{L}(D = d; \mu_{M_q} = m, \Sigma_{M_q} = R, \Sigma_q = S) = P_{m,R,S}(D = d) \tag{163}$$

$$(\hat{\mu}_{M_q}^{MLE}, \hat{\Sigma}_{M_q}^{MLE}, \hat{\Sigma}_q^{MLE}) = \underset{m,R,S}{\operatorname{argmax}} \mathcal{L}(D = d; \mu_{M_q} = m, \Sigma_{M_q} = R, \Sigma_q = S) \tag{164}$$

By solving Equation (164) we yield point estimates $\hat{\mu}_{M_q}^{MLE}, \hat{\Sigma}_{M_q}^{MLE}$ and $\hat{\Sigma}_q^{MLE}$ for μ_{M_q}, Σ_{M_q} and Σ_q. For notational convenience we use:

$$\theta = (\mu_{M_q}, \Sigma_{M_q}, \Sigma_q) \tag{165}$$

$$\hat{\theta} = (\hat{\mu}_{M_q}, \hat{\Sigma}_{M_q}, \hat{\Sigma}_q) \tag{166}$$

$$\hat{\theta}^{MLE} = (\hat{\mu}_{M_q}^{MLE}, \hat{\Sigma}_{M_q}^{MLE}, \hat{\Sigma}_q^{MLE}) \tag{167}$$

We use the following equations to construct the likelihood expression:

$$\mathcal{L}(\hat{\theta}; D = d) = P_{\hat{\theta}}(D = d) = \prod_m P_{\hat{\theta}}(D_m = d_m) \tag{168}$$

$$P_{\hat{\theta}}(D_m = k) \approx (\beta_k - \beta_{k-1})\mathcal{N}(\hat{\mu}_{dS_m}; \mu_k, \sigma_{D_k}^2 + \hat{\sigma}_{dS_m}^2) \text{ for } k = 2, ..., K-1 \tag{169}$$

$$P_{\hat{\theta}}(D_m = 1) = \Phi\left(\frac{\beta_1 - \hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right) \tag{170}$$

$$P_{\hat{\theta}}(D_m = K) = 1 - \Phi\left(\frac{\beta_{K-1} - \hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right) \tag{171}$$

In these Equations we use m as the match index and we define:

$$\hat{\mu}_{dS_m} = \delta_m \hat{\mu}_{M_q} \tag{172}$$

$$\hat{\sigma}_{dS_m}^2 = \delta_m(\hat{\Sigma_q} + \hat{\Sigma}_{M_q})\delta_m^T \tag{173}$$

By using Equations (136) and (140) we can write down the likelihood explicitly as:

$$P_{\hat{\theta}}(D = d) = \prod_m P_{\hat{\theta}}(D_m = d_m) \tag{174}$$

$$\begin{aligned} &= \prod_m \Phi\left(\frac{\beta_1 - \hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right)^{\mathbb{1}(d_m=1)} \\ &\times \left(1 - \Phi\left(\frac{\beta_{K-1} - \hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right)\right)^{\mathbb{1}(d_m=K)} \\ &\times \prod_{j=2}^{K-1} \left[\left((\beta_j - \beta_{j-1})\mathcal{N}(\hat{\mu}_{dS_m}; \mu_{D_j}, \sigma_{D_j}^2 + \hat{\sigma}_{dS_m}^2)\right)^{\mathbb{1}(d_m=j)}\right] \end{aligned} \tag{175}$$

We take the logarithm of this expression to yield the following log-likelihood expression:

$$\begin{aligned}\ln P_{\hat{\theta}}(D=d) \propto & \sum_m \mathbb{1}(D_m=1)\ln\Phi\left(\frac{\beta_1-\hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right) \\ & + \mathbb{1}(D_m=K)\ln\left(1-\Phi\left(\frac{\beta_{K-1}-\hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right)\right) \qquad (176) \\ & + \sum_{j=2}^{K-1}\mathbb{1}(D_m=j)\left(\ln(\beta_k-\beta_{k-1})+\ln\mathcal{N}(\hat{\mu}_{dS_m};\mu_{D_k},\sigma_{D_k}^2+\hat{\sigma}_{dS_m}^2)\right) \\ = & \sum_m \mathbb{1}(D_m=1)\ln\Phi\left(\frac{\beta_1-\hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right) \\ & + \mathbb{1}(D_m=K)\ln\left(1-\Phi\left(\frac{\beta_{K-1}-\hat{\mu}_{dS_m}}{\hat{\sigma}_{dS_m}}\right)\right) \qquad (177) \\ & + \frac{1}{2}\sum_{j=2}^{K-1}\mathbb{1}(D_m=j)\frac{(\hat{\mu}_{dS_m}-\mu_{D_k})^2}{\sqrt{\sigma_{D_k}^2+\hat{\sigma}_{dS_m}^2}}+Constants\end{aligned}$$

We can now find $\hat{\mu}_{M_q}^{MLE},\hat{\Sigma}_{M_q}^{MLE},\hat{\Sigma}_q^{MLE}=\underset{\mu_{M_q},\Sigma_{M_q},\Sigma_q}{\operatorname{argmax}}\ln\mathcal{L}(\mu_{M_q},\Sigma_{M_q},\Sigma_q;D)$, as the maximum likelihood estimators for this problem. Unfortunately, direct optimization of this equation does not work, because the dimensionality of the feasible space is too large. Σ_q has $O(N_PN_Q)^2$ elements, which is too expensive to optimize numerically. Even if we would only consider active players within a football match we would have $N_P=26$, we could choose $N_Q=2$ and even this *small problem size* can't be solved efficiently.
Numerical feasibility is not the only issue with this approach. There is no way to distinguish the influences by Σ_{M_q} and Σ_q because their influence on $\sigma_{dS_m}^2$ is identical, as we can see in Equation (173). Another downside of this approach is that for it to be feasible we would need to assume that μ_{Mu_q}, Σ_{M_q}, and Σ_q are constant over the whole dataset, but it is much more realistic for the player ratings to vary over time. In the next subsections, we develop methodologies that are much more efficient direct inference methods, that can process batches and large amounts of players.

5.2 Ordinary least squares

Ordinary least squares (OLS) is a widely used estimation method for linear problems (Rao, 1973). The idea is that we want to minimize the Euclidian norm of the error vector, effectively minimizing the sum of squared differences between our predictions and real observations. We assume that the player inconsistencies are heteroskedastic, which means that $\Sigma_q=\sigma^2I$ where $\sigma>0$. We get the following, unbiased OLS-estimator for μ_{M_q}:

$$\begin{aligned}Q &\sim \mathcal{N}(M_q,\Sigma_q) && (178)\\ M_q &\sim \mathcal{N}\left(\mu_{M_q},\Sigma_{M_q}\right) && (179)\\ dS_m &= \delta_m Q && (180)\\ dS &= \delta Q && (181)\\ dS_m^O &= \delta_m\hat{M}_q+\epsilon_m && (182)\\ dS^O &= \delta\hat{M}_q+\epsilon && (183)\\ \epsilon &\sim \mathcal{N}\left(0,\Sigma_{dS}\right)=\mathcal{N}(0,\delta\Sigma_q\delta^T\circ I)=\mathcal{N}(0,\sigma^2(\delta\delta^T\circ I)) && (184)\\ \hat{M}_q &= \mathcal{N}(\hat{\mu}_{M_q},\hat{\Sigma}_{M_q}) && (185)\\ \hat{\mu}_{M_q}^{OLS} &= \underset{m}{\operatorname{argmin}}(dS-\delta m)^T(dS-\delta m) && (186)\\ &= \underset{m}{\operatorname{argmin}}\,\|dS-\delta m\|_2 && (187)\\ &= (\delta^T\delta)^{-1}\delta^T dS && (188)\end{aligned}$$

This estimator only exists under the assumption that δ has full row rank and therefore $(\delta^T\delta)^{-1}$ exists. The estimator is unbiased because:

$$E[\hat{\mu}_{M_q}^{OLS}] = E[(\delta^T\delta)^{-1}\delta^T dS] \tag{189}$$
$$= (\delta^T\delta)^{-1}\delta^T E[dS] \tag{190}$$
$$= (\delta^T\delta)^{-1}\delta^T\delta\mu_{M_q} = \mu_{M_q} \tag{191}$$

The variance of this estimator is:

$$\text{Var}\left(\hat{\mu}_{M_q}^{OLS}\right) = \text{Var}\left(\left(\delta^T\delta\right)^{-1}\delta^T dS\right) \tag{192}$$
$$= \left(\delta^T\delta\right)^{-1}\delta^T\text{Var}(dS)\delta\left(\delta^T\delta\right)^{-1} \tag{193}$$
$$= \left(\delta^T\delta\right)^{-1}\delta^T\Sigma_{dS}\delta\left(\delta^T\delta\right)^{-1} \tag{194}$$

We use matrix operations from (Petersen & Pedersen, 2012) to find estimates for the other parameters:

$$\Sigma_{dS} = \sigma^2(\delta\delta^T \circ I) \tag{195}$$
$$\sigma_{dS_m}^2 = \sigma^2\sum_{j=1}^{N_{PQ}}\delta_{m,j}^2 \tag{196}$$
$$\epsilon = dS - \hat{dS} \tag{197}$$
$$= dS - \delta\hat{\mu}_{M_q} \tag{198}$$
$$= dS - \delta(\delta^T\delta)^{-1}\delta^T dS \tag{199}$$
$$= (I - \delta(\delta^T\delta)^{-1}\delta^T)dS \tag{200}$$
$$\epsilon^T\epsilon = dS^T(I - \delta(\delta^T\delta)^{-1}\delta^T)^T(I - \delta(\delta^T\delta)^{-1}\delta^T)dS \tag{201}$$
$$= dS^T(I - \delta(\delta^T\delta)^{-1}\delta^T)dS \tag{202}$$
$$E[\epsilon^T\epsilon] = E[dS^T(I - \delta(\delta^T\delta)^{-1}\delta^T)dS] \tag{203}$$
$$= E[(\delta\mu_q + \epsilon)^T(I - \delta(\delta^T\delta)^{-1}\delta^T)(\delta\mu_q + \epsilon)] \tag{204}$$
$$= E[\epsilon^T(I - \delta(\delta^T\delta)^{-1}\delta^T)\epsilon] \tag{205}$$
$$= Tr((I - \delta(\delta^T\delta)^{-1}\delta^T)\Sigma_{dS}) \tag{206}$$
$$= Tr(\Sigma_{dS}) - Tr(\delta(\delta^T\delta)^{-1}\delta^T\Sigma_{dS}) \tag{207}$$
$$= \sigma^2 Tr(\delta^T\delta) - \sigma^2 Tr(\delta(\delta^T\delta)^{-1}\delta^T(\delta\delta^T \circ I)) \tag{208}$$
$$(\hat{\sigma}^{OLS})^2 = \frac{\epsilon^T\epsilon}{Tr(\delta^T\delta) - Tr(\delta(\delta^T\delta)^{-1}\delta^T(\delta\delta^T \circ I))} \tag{209}$$
$$\hat{\Sigma}_q^{OLS} = (\hat{\sigma}^{OLS})^2 I \tag{210}$$
$$\hat{\Sigma}_{dS}^{OLS} = (\hat{\sigma}^{OLS})^2(\delta^T\delta \circ I) \tag{211}$$
$$\hat{\Sigma}_{M_q}^{OLS} = \text{Var}((\delta^T\delta)^{-1}\delta^T dS) \tag{212}$$
$$= (\delta^T\delta)^{-1}\delta^T\ \hat{\Sigma}_{dS}^{OLS}\ \delta(\delta^T\delta)^{-T} \tag{213}$$
$$= \hat{\sigma}^2(\delta^T\delta)^{-1}\delta^T(\delta\delta^T \circ I)\delta(\delta^T\delta)^{-1} \tag{214}$$

Note that $(\hat{\sigma}^{OLS})^2$ is unbiased, as equation (209) in expectation is equivalent to Equation (208). In some cases we have that $\delta_{*,i}\delta_{*,i}^T \approx c\ \forall_i$, this gives us that $(\delta^T\delta \circ I) \approx c \cdot I$, therefore we yield:

$$E[\hat{\epsilon}^T\epsilon] \approx \sigma^2 Tr(\delta^T\delta)(1 - \frac{cN_{PQ}}{N_{KM}}) \tag{215}$$
$$(\hat{\sigma}^{OLS})^2 \approx \frac{\epsilon^T\epsilon N_{KM}}{Tr(\delta^T\delta)(N_{KM} - cN_{PQ})} \tag{216}$$
$$\hat{\Sigma}_{M_q} \approx (\hat{\sigma}^{OLS})^2 \cdot c \cdot (\delta^T\delta)^{-1} \tag{217}$$

5.3 Generalized least squares

An extension to the OLS estimation is Generalized Least Squares Estimator (GLS). The method utilizes a matrix Ω, which must be an invertible matrix that we can choose freely. While in OLS all samples carry equal weight, in GLS the Ω matrix can be seen as the weighting applied to our observations. If Σ_{dS} is known, in general $\Omega = \Sigma_{dS}$ is chosen as this minimizes the variance of the estimator. By applying the GLS method we find the estimator by:

$$\hat{\mu}_{M_q}^{GLS} = \underset{m}{\operatorname{argmin}} (dS - \delta m)^T \Omega (dS - \delta m) \tag{218}$$

$$= \underset{m}{\operatorname{argmin}} \|dS - \delta m\|_\Omega \tag{219}$$

$$= (\delta^T \Omega \delta)^{-1} \delta^T \Omega dS \tag{220}$$

This estimator is also unbiased, as we have that:

$$E[\hat{\mu}_{M_q}^{GLS}] = E[(\delta^T \Omega \delta)^{-1} \delta^T \Omega dS] \tag{221}$$

$$= (\delta^T \Omega \delta)^{-1} \delta^T \Omega E[dS] \tag{222}$$

$$= (\delta^T \Omega \delta)^{-1} \delta^T \Omega \delta \mu_{M_q} \tag{223}$$

$$= \mu_{M_q} \tag{224}$$

Analogous to the equations in Section 5.2, under the assumption that $\Sigma_q = \sigma^2 I$, we can now derive the estimates for the other parameters.

$$(\hat{\sigma}^{GLS})^2 = \frac{\epsilon^T \epsilon}{Tr(\delta^T \Omega \delta) - Tr(\delta (\delta^T \Omega \delta)^{-1} \delta^T \Omega (\delta \delta^T \circ I))} \tag{225}$$

$$\hat{\Sigma}_q^{GLS} = (\hat{\sigma}^{GLS})^2 I \tag{226}$$

$$\hat{\Sigma}_{dS}^{GLS} = (\hat{\sigma}^{GLS})^2 (\delta \delta^T \circ I) \tag{227}$$

$$\hat{\Sigma}_{M_q}^{GLS} = \operatorname{Var}((\delta^T \Omega \delta)^{-1} \delta^T \Omega dS) \tag{228}$$

$$= (\hat{\sigma}^{GLS})^2 (\delta^T \Omega \delta)^{-1} \tag{229}$$

5.4 Regularized estimates

The OLS and GLS methodology applied in previous sections have quite a few drawbacks. Even though our estimators were unbiased, their variance can be very high. Also, we required $\delta^T \delta$ (OLS) or $\delta^T \Omega \delta$ (GLS) to be an invertible matrix which is not always the case. In some scenario's, just like in ours, this requires a lot of data. A technique that is often used to overcome the previously mentioned problems is regularization of the estimator. We discuss a method called Tikhonov regularization, more often referred to as ridge regression (RR) (Hoerl & Kennard, 1970). This method effectively damps the effect of the data on our predictions, regularizing the estimators we yield. The idea is that we want to achieve informative parameters, while not over-fitting to the historical data. In the approach we have that:

$$\hat{\mu}_{M_q}^{RR} = \underset{m}{\operatorname{argmin}} \|dS - \delta m\|_2 + \lambda \|m\|_2 \quad \text{where } \lambda > 0 \tag{230}$$

$$\hat{\mu}_{M_q}^{RR} = (\delta^T \delta + \lambda I)^{-1} \delta^T dS \tag{231}$$

$$\hat{\Sigma}_{M_q}^{RR} = \operatorname{Var}(\hat{\mu}_{M_q}^{RR}) \tag{232}$$

$$= (\delta^T \delta + \lambda I)^{-1} \delta^T \Sigma_{dS} \delta (\delta^T \delta + \lambda I)^{-T} \tag{233}$$

Here we choose a parameter $\lambda > 0$. Unfortunately, this estimator is biased, as we have that:

$$E[\hat{\mu}_{M_q}] = (\delta^T \delta + \lambda I)^{-1} \delta^T \delta E[M_q] \tag{234}$$

$$= \left[I - \lambda ((\delta^T \delta + \lambda I)^{-1})\right] \mu_q = \mu_{M_q} - \lambda (\delta^T \delta + \lambda I)^{-1} \mu_{M_q} \tag{235}$$

The fact that our estimator is biased is obviously a disadvantage of this method. On the other hand, this method has two advantages. Firstly; we do not require $\delta^T \delta$ to be invertible, as the inverse of $(\delta^T \delta + \lambda I), \lambda > 0$ always exists. Secondly, even though we introduce bias, the mean squared error of our estimator can decrease significantly. The choice of λ depends on the specific application. In general, it holds that larger (smaller) λ leads to a larger (smaller) bias and a smaller (larger) variance.

We will show that rigde regression, with appropriately chosen λ is equivalent to applying MLE in a Bayesian context. We will assume that $\text{Var}(dS) = \sigma_{dS}^2 I$ and we take a prior distribution over $M_q \sim \mathcal{N}(0, \sigma_{t_0}^2 I)$. We follow the standard maximum likelihood estimation procedure;

$$\mathcal{L}(dS = ds; M_q = m) \propto p_{dS}(ds|M_q = m) P_{M_q}(m) \tag{236}$$
$$= \mathcal{N}(ds; \delta m, \sigma_{dS}^2 I)\mathcal{N}(m; 0, \sigma_{t_0}^2 I) \tag{237}$$
$$\propto \exp(-\sigma_{dS}^{-2}(ds - \delta m)(ds - \delta m)^T - \sigma_{t_0}^{-2} m m^T) \tag{238}$$
$$\ln \mathcal{L}(dS = ds|M_q = m) \propto -(ds - \delta m)(ds - \delta m)^T - \left(\frac{\sigma_{dS}}{\sigma_{t_0}} m\right)\left(\frac{\sigma_{dS}}{\sigma_{t_0}} m\right)^T \tag{239}$$
$$\hat{\mu}_q^{BRR} = \underset{m}{\text{argmin}} \|ds - \delta m\|_2 + \frac{\sigma_{dS}^2}{\sigma_{t_0}^2}\|m\|_2 \tag{240}$$

We see that Ridge Regression with the choice $\lambda = \frac{\sigma_{dS}^2}{\sigma_{t_0}^2}$ is therefore equivalent to this method.

5.5 Batch inference

In some cases we would like to know the evolution of our estimate over time; we can achieve this by processing all the data in batches rather than in a single run. Batch processing is very intuitive when applying a Bayesian method, as we will do in Section 5.6. For such estimators, we simply use the posterior as the prior for a next batch. For our OLS estimator of the mean quality, we get the following equations in the case that we use two batches:

$$\begin{bmatrix} dS_{t_1} \\ dS_{t_2} \end{bmatrix} = \begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix} M_q + \begin{bmatrix} \epsilon_{t_1} \\ \epsilon_{t_2} \end{bmatrix} \tag{241}$$
$$\begin{bmatrix} \epsilon_{t_1} \\ \epsilon_{t_2} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \delta_{t_1}\Sigma_q\delta_{t_1}^T \circ I & 0 \\ 0 & \delta_{t_2}\Sigma_q\delta_{t_2}^T \circ I \end{bmatrix}\right) \tag{242}$$
$$\hat{\mu}_{M_q}^{OLS} = \left(\begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix}^T \begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix}\right)^{-1} \begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix}^T \begin{bmatrix} dS_{t_1} \\ dS_{t_2} \end{bmatrix} \tag{243}$$
$$= \left(\delta_{t_1}^T\delta_{t_1} + \delta_{t_2}^T\delta_{t_2}\right)^{-1}\left[\delta_{t_1}^T dS_{t_1} + \delta_{t_2}^T dS_{t_2}\right] \tag{244}$$

Here we require that $\delta_{t_1}^T\delta_{t_1}$ is invertible, as the first batch needs to yield results by itself. This will guarantee that $\left(\delta_{t_1}^T\delta_{t_1} + \delta_{t_2}^T\delta_{t_2}\right)$ is invertible, because this matrix is symmetric and positive definite. Both $\delta_{t_1}^T\delta_{t_1}$ and $\delta_{t_2}^T\delta_{t_2}$ are symmetric and PSD. Due to the fact that $\delta_{t_1}^T\delta_{t_1}$ is invertible and symmetric, it must be positive definite. The sum of a symmetric PD matrix and a symmetric PSD matrix is a symmetric PD matrix.
The idea is now that if we want to apply batch processing, all processed batches are aggregated in t_1, and the new batch is t_2. This also means that the inverse of $\delta_{t_1}^T\delta_{t_1}$ is readily available, from the previous iteration. We note that the computationally most expensive step during each iteration is the calculation of the inverse of $\left(\delta_{t_1}^T\delta_{t_1} + \delta_{t_2}^T\delta_{t_2}\right)$. Once we have this inverse the estimators of $\hat{\Sigma}_{M_q}$ and $\hat{\Sigma}_q$ can be calculated efficiently with Equations (209), (210) and (214).
We get very similar equations for the GLS methodology:

$$\hat{\mu}_{M_q}^{GLS} = \left(\begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix}^T \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix}\right)^{-1} \begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix}^T \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \begin{bmatrix} dS_{t_1} \\ dS_{t_2} \end{bmatrix} \tag{245}$$
$$= \left(\delta_{t_1}^T\Omega_{11}\delta_{t_1} + \delta_{t_1}^T\Omega_{12}\delta_{t_2} + \delta_{t_2}^T\Omega_{21}\delta_{t_1} + \delta_{t_2}^T\Omega_{22}\delta_{t_2}\right)^{-1} \begin{bmatrix} \delta_{t_1} \\ \delta_{t_2} \end{bmatrix}^T \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \begin{bmatrix} dS_{t_1} \\ dS_{t_2} \end{bmatrix} \tag{246}$$

In this case the structure of Ω can be chosen freely, though it must be an invertible matrix. Again the computationally challenging step is finding the inverse of a large matrix:

$$\left(\delta_{t_1}^T\Omega_{11}\delta_{t_1} + \delta_{t_1}^T\Omega_{12}\delta_{t_2} + \delta_{t_2}^T\Omega_{21}\delta_{t_1} + \delta_{t_2}^T\Omega_{22}\delta_{t_2}\right) \tag{247}$$

Once we have this, the estimators of $\hat{\Sigma}_{M_q}$ and $\hat{\Sigma}_q$ can be calculated efficiently with Equations 225 to 229.

5.5.1 Woodbury matrix identity

We found that the Woodbury matrix identity was very useful to apply for computational efficiency (Press *et al.* , 1992, Woodbury Formula Sec. 2.7.3):

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1} \tag{248}$$

$$A \in \mathbb{R}^{S\times S}, C \in \mathbb{R}^{T\times T}, U \in \mathbb{R}^{S\times T}, V \in \mathbb{R}^{T\times S} \tag{249}$$

This method assumes that A and C are invertible, and the inverse of A is known (or easy to calculate). Rather than inverting $A+UCV$, a $S\times S$ matrix, we only need to invert C, a $T\times T$ matrix and perform some matrix multiplications. If $S \gg T$, applying this identity can save a lot of computation time. For the application in OLS we have:

$$A = \delta_{t_1}^T\delta_{t_1}, U = \delta_{t_2}^T, C = I, V = \delta_{t_2} \tag{250}$$

For the special case that $rank(\delta_{t_2}^T\delta_{t_2}) = 1$ and $(\delta_{t_1}^T\delta_{t_1}^T)^{-1}$ is known from the previous iteration we get the following equations, by using (Press *et al.* , 1992, Sherman-Morrison formula Sec. 2.7.1):

$$(G+H)^{-1} = G^{-1} - \frac{1}{1+Tr(G^{-1}H)}G^{-1}HG^{-1} \tag{251}$$

$$G = \delta_{t_1}^T\delta_{t_1} \tag{252}$$

$$H = \delta_{t_2}^T\delta_{t_2} \tag{253}$$

This procedure can be very useful if we want to apply online updates, after each match. It is much more efficient because no large inverses of matrices (except during the first iteration), have to be computed.
We can also apply the Woodbury matrix identity to the GLS batch inference. The appropriate way depends on the choice for the matrix Ω. If we have $\Omega_{12} = \Omega_{21} = 0$, we can use:

$$A = \delta_{t_1}^T\Omega_{11}\delta_{t_1}, U = \delta_{t_2}^T, C = \Omega_{22}, V = \delta_{t_2} \tag{254}$$

This specific choice of Ω makes sense in our case, as we assume that $\mathrm{Cov}(dS_{t_1}, dS_{t_2}) = 0$. Under the condition that $rank(\delta_{t_2}^T\Omega_{22}\delta_{t_2}) = 1$, we can apply Equation 251 with:

$$G = \delta_{t_1}^T\Omega_{11}\delta_{t_1} \tag{255}$$

$$H = \delta_{t_2}^T\Omega_{22}\delta_{t_2} \tag{256}$$

5.6 Conditional Gaussian inference

In this section we will consider a Bayesian estimation method; we will use a prior and calculate the posterior probability distribution conditioned on the available data. As stated in Equation (162), the estimator that minimizes the MSE is the conditional expectation of the variable that we want to estimate. In general, the conditional distribution cannot be calculated analytically, but because our distributions are Multivariate Gaussian we can find the conditional distributions analytically. We describe the method, taken from (Bishop, 2006) in Appendix C.4.
To apply this method we require a prior distribution over $M_q \sim \mathcal{N}(\mu_{M_q}^0, \Sigma_q^0)$ and an estimate for Σ_q. In theory this inference equations of this method will work for any choice of $\mu_{M_q}^0, \Sigma_q^0$ and Σ_q. A separate method can be used to determine a value of Σ_q, but we will take $\Sigma_q = (\sigma_q^0)^2 I$, assuming that all players have the same performance randomness. The parameters $\mu_{M_q}^0$ and Σ_q^0 are prior parameters, for unknown players we take:

$$\mu_{M_q}^0 = m_0\vec{\mathbb{1}} \tag{257}$$

$$\Sigma_{M_q}^0 = (\sigma_{M_q}^0)^2 I \tag{258}$$

Here $\vec{\mathbb{1}}$ is a column vector filled with ones. This means that each player quality will start out with mean m_0 and uncertainty $(\sigma_{M_q}^0)^2$.
We have the following equations:

$$p_Q(q) = \mathcal{N}(q; M_q, \Sigma_q) \tag{259}$$

$$p_{M_q}(x) = \mathcal{N}(x; \mu_{M_q}^0, \Sigma_{M_q}^0) \tag{260}$$

$$p_{dS}(ds|M_q = x) = \mathcal{N}(y; \delta x, \Sigma_{dS}) = \mathcal{N}(y; \delta x, \delta\Sigma_q\delta^T \circ I) \tag{261}$$

$$p_{dS}(y) = \mathcal{N}(y; \delta\mu_{M_q}^0, \delta(\Sigma_{M_q}^0 + \Sigma_q)\delta^T) \tag{262}$$

As described in Appendix C.4 we yield:

$$p_{M_q}(x|dS = y) = \mathcal{N}(x; \hat{\mu}_{M_q}, \hat{\Sigma}_{M_q}) \tag{263}$$

$$\hat{\Sigma}_{M_q} = \left((\Sigma^0_{M_q})^{-1} + \delta^T(\Sigma_{dS})^{-1}\delta\right)^{-1} \tag{264}$$

$$= \left((\Sigma^0_{M_q})^{-1} + \delta^T(\delta\Sigma_q\delta^T \circ I)^{-1}\delta\right)^{-1} \tag{265}$$

$$= \left((\Sigma^0_{M_q})^{-1} + (\sigma^0_q)^{-2}\delta^T(\delta\delta^T \circ I)^{-1}\delta\right)^{-1} \tag{266}$$

$$\hat{\mu}_{M_q} = \hat{\Sigma}_{M_q}(\delta^T\Sigma^{-1}_{dS}y + (\Sigma^0_{M_q})^{-1}\mu^0_{M_q}) \tag{267}$$

The idea is to apply this method in batches, using the posterior of each batch as the prior for the next batch. This also allows us to keep track of incremental changes in the estimators, so we can see player quality evolving over time. We prove that batch processing is equivalent to processing all the data in a single iteration in Section 5.6.1.
The bias of the estimator $\hat{\mu}_{M_q}$ and the norm the estimator of Σ_{M_q} decrease each iteration. This shows that the method converges, and is asymptotically unbiased. We prove this in Section C.5 of the appendix.

5.6.1 Batch processing

In this section we will show that application of the conditional Gaussian Inference method in batches is equivalent to inference over the complete dataset.
We will assume that our dataset can be separated in N_B non-overlapping batches. We use that:

$$\delta = \begin{bmatrix} \delta^{(1)} \\ \delta^{(2)} \\ ... \\ \delta^{(N_B)} \end{bmatrix}, dS = \begin{bmatrix} dS^{(1)} \\ dS^{(2)} \\ ... \\ dS^{(N_B)} \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_{N_B} \end{bmatrix} \tag{268}$$

$$\Sigma_{dS} = \begin{bmatrix} \Sigma^{(1)}_{dS} & 0 & ... & 0 \\ 0 & \Sigma^{(2)}_{dS} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \Sigma^{(N_B)}_{y} \end{bmatrix} \tag{269}$$

First, we apply the model to two batches:

$$p_{M_q}(x|dS^{(1)} = y_1, dS^{(2)} = y_2) = \mathcal{N}(x; \mu_{M_q|y_1,y_2}, \Sigma_{M_q|y_1,y_2}) \tag{270}$$

$$\mu_{M_q|y_1,y_2} = \Sigma_{M_q|y_1,y_2}\left(\begin{bmatrix}\delta^{(1)} \\ \delta^{(2)}\end{bmatrix}^T \begin{bmatrix}\Sigma^{(1)}_{dS} & 0 \\ 0 & \Sigma^{(2)}_{dS}\end{bmatrix}^{-1} \begin{bmatrix}y_1 \\ y_2\end{bmatrix} + (\Sigma^0_{M_q})^{-1}\mu^0_{M_q}\right) \tag{271}$$

$$= \Sigma_{M_q|y_1,y_2}\left((\delta^{(1)})^T(\Sigma^{(1)}_{dS})^{-1}y_1 + (\delta^{(2)})^T(\Sigma^{(2)}_{dS})^{-1}y_2 + (\Sigma^0_{M_q})^{-1}\mu^0_{M_q}\right) \tag{272}$$

$$\Sigma_{M_q|y_1,y_2} = \left((\Sigma^0_{M_q})^{-1} + \begin{bmatrix}\delta^{(1)} \\ \delta^{(2)}\end{bmatrix}^T \begin{bmatrix}\Sigma^{(1)}_{dS} & 0 \\ 0 & \Sigma^{(2)}_{dS}\end{bmatrix}^{-1} \begin{bmatrix}\delta^{(1)} \\ \delta^{(2)}\end{bmatrix}\right)^{-1} \tag{273}$$

$$= \left((\Sigma^0_{M_q})^{-1} + (\delta^{(1)})^T(\Sigma^{(1)}_{dS})^{-1}\delta^{(1)} + (\delta^{(2)})^T(\Sigma^{(2)}_{dS})^{-1}\delta^{(2)}\right)^{-1} \tag{274}$$

We define $Z = (M_q|dS^{(1)} = y_1)$, which is the conditional random variable we would get if we would condition only on the first batch. We get:

$$p_{M_q}(x|dS^{(1)} = y_1) = p_{M_q|y_1}(x) = p_Z(x) = \mathcal{N}(x; \mu_{M_q|y_1}, \Sigma_{M_q|y_1}) \tag{275}$$

$$\mu_{M_q|y_1} = \mu_Z = \Sigma_{M_q|y_1}\left((\delta^{(1)})^T\Sigma^{(1)}_{dS}y_1 + \Sigma^0_{M_q}\mu^0_{M_q}\right) \tag{276}$$

$$\Sigma_{M_q|y_1} = \Sigma_Z = \left(\Sigma^0_{M_q} + (\delta^{(1)})^T\Sigma^{(1)}_{dS}\delta^{(1)}\right)^{-1} \tag{277}$$

Now we can apply the same model to the random variable Z with the observations $dS^{(2)} = y_2$, to extract $(Z|dS^{(2)} = y_2)$. As expected, we get that $(Z|dS^{(2)} = y_2) = (M_q|dS^{(1)} = y_1, dS^{(2)} = y_2)$,

because $dS^{(1)}$ and $dS^{(2)}$ are independent. We get:

$$p_Z(x|dS^{(2)} = y_2) = p_{Z|y_2}(x) = \mathcal{N}(x; \mu_{Z|y_2}, \Sigma_{Z|y_2}) \tag{278}$$

$$\mu_{Z|y_2} = \Sigma_{Z|y_2}\left((\delta^{(2)})^T(\Sigma^{(2)}_{dS})^{-1}y_2 + (\Sigma_Z)^{-1}\mu_Z\right) \tag{279}$$

$$= \Sigma_{Z|y_2}\left((\delta^{(2)})^T(\Sigma^{(2)}_{dS})^{-1}y_2 + (\delta^{(1)})^T(\Sigma^{(1)}_{dS})^{-1}y_1 + (\Sigma^0_{M_q})^{-1}\mu^0_{M_q}\right) \tag{280}$$

$$\Sigma_{Z|y_2} = \left(\Sigma_Z^{-1} + (\delta^{(2)})^T(\Sigma^{(2)}_{dS})^{-1}\delta^{(2)}\right)^{-1} \tag{281}$$

$$= \left((\Sigma^0_{M_q})^{-1} + (\delta^{(1)})^T(\Sigma^{(1)}_{dS})^{-1}\delta^{(1)} + (\delta^{(2)})^T(\Sigma^{(2)}_{dS})^{-1}\delta^{(2)}\right)^{-1} \tag{282}$$

This means that sequential application of the algorithm on two independent sets of results is the same as application of the algorithm on all the results at once. This property can be extended to hold for any partition in batches, under the conditions that batches are non-overlapping and the union of all batches contains all the results.
The result we yield is not surprising; effectively when conditioning on two independent random variables $dS^{(1)}$ and $dS^{(2)}$ we are projecting our random variable M_q on two independent subspaces. We use the initial prior Equations from (257) and (258), and get the following equations for the N^{th} batch:

$$\hat{\mu}^{(N)}_{M_q} = \hat{\Sigma}^{(N)}_{M_q}\left((\delta^{(N)})^T\hat{\Sigma}^{(N)}_{dS}y_N + \hat{\Sigma}^{N-1}_{M_q}\hat{\mu}^{N-1}_{M_q}\right) \tag{283}$$

$$\hat{\Sigma}^{(N)}_{M_q} = \left(\hat{\Sigma}^{(N-1)}_{M_q} + (\delta^{(N)})^T\hat{\Sigma}^{(N)}_{dS}\delta^{(N)}\right)^{-1} \tag{284}$$

The most expensive step during each batch is calculating the inverse of $\left(\hat{\Sigma}^{(N-1)}_{M_q} + (\delta^{(N)})^T\hat{\Sigma}^{(N)}_{dS}\delta^{(N)}\right)$. We can apply the Woodbury inverse from Equation (248), keeping in mind that we have $\left(\hat{\Sigma}^{(N-1)}_{M_q}\right)^{-1}$ readily available from the previous iteration, we use:

$$A = \Sigma^{N-1}_{M_q}, U = (\delta^{(N)})^T, C = \Sigma^{(N)}_{dS}, V = \delta^{(N)} \tag{285}$$

This gives us the following update equation:

$$\hat{\Sigma}^{(N)}_{M_q} = \left(\hat{\Sigma}^{(N-1)}_{M_q}\right)^{-1} - \left(\hat{\Sigma}^{(N-1)}_{M_q}\right)^{-1}\left(\delta^{(N)}\right)^T\left(\left(\hat{\Sigma}^{(N)}_{dS}\right)^{-1} + \Sigma^{(N)}_{dS}\left(\hat{\Sigma}^{(N-1)}_{M_q}\right)^{-1}\left(\delta^{(N)}\right)^T\right)^{-1}\delta^{(N)}\left(\hat{\Sigma}^{(N-1)}_{M_q}\right)^{-1} \tag{286}$$

Using this equation gives a significant improvement of computation time.

5.7 Estimation using bookmaker predictions

As discussed in Section 4.4 we can use the probabilities provided by bookmakers to infer the ratings of the players participating in the match. We define the probabilities implied by the bookmaker by $P_B(D = d)$. There are two main approaches we can take. The first approach is discussed in Section 5.7.1 and it uses the bookmaker implied probabilities $P(D = d|q \sim \mathcal{N}(M_q, \Sigma_q)))$ within an approach similar to MLE, to find point estimators for M_q and Σ_q. In Section 5.7.2 we discuss the second approach where we infer the strength difference between teams, implied by bookmakers. We can use this strength difference as an additional observation next to other KPI-outcomes of a match. Both methods require the usage of the Kullback–Leibler divergence. This is a measure of the difference between two probability distributions. It is defined for discrete probability distributions by the following formula:

$$\mathbb{D}_{KL}(P||Q) = \sum_x P(x)\log\left(\frac{P(x)}{Q(x)}\right) \tag{287}$$

Note that the Kullback-Leibler Divergence is non-associative; we have that:

$$\mathbb{D}_{KL}(P||Q) \neq \mathbb{D}_{KL}(Q||P) \tag{288}$$

5.7.1 Bookmaker maximum likelihood

In the first approach, we will find estimators for M_q and Σ_q such that the predictions of the bookmakers correspond to predictions by our model. We measure the correspondence of the bookmaker's and our model's outcome probability distribution by calculating their Kullback-Leibler divergence. We get the following:

$$\hat{M}_q^{BMLE}, \hat{\Sigma}_q^{BMLE} = \underset{M_q, \Sigma_q}{\operatorname{argmin}} \mathbb{D}_{KL}(P(D = d | q \sim \mathcal{N}(M_q, \Sigma_q)) || P_B(D = d)) \tag{289}$$

Effectively we are finding estimates $\hat{M}_q^{BMLE}, \hat{\Sigma}_q^{BMLE}$, such that we minimize the KullBack-Leibler divergence between the predictions of the model and the predictions of the bookmakers. The problem in equation (289) is not trivial to solve.

5.7.2 Bookmaker implied strength difference

In this subsection, we propose the procedure whereby we calculate the implied strength differences for each match to develop estimators for μ_{M_q}, Σ_{M_q} and Σ_q. We will represent our estimate of the strength difference implied by the bookmaker for a single match m as dS_m^B. We get:

$$(\hat{\mu}_{dS_m^B}, \hat{\sigma}^2_{dS_m^B}) = \underset{s, r^2}{\operatorname{argmin}} \mathbb{D}_{KL}\left(P(D_m = d_m | dS_m \sim \mathcal{N}(s, r^2)) \middle|\middle| P_B(D_m = d_m)\right) \quad \forall \text{ matches } m \tag{290}$$

Once we have the bookmaker implied strength differences for each match, there are two approaches. One approach is to directly maximize the likelihood of bookmaker implied strengths. We get $\hat{M}_q^{BIM}$ and $\hat{\Sigma}_q^{BIM}$ as follows:

$$(\hat{\mu}_{M_q}^{BIM}, \hat{\Sigma}_{M_q}^{BIM}) = \underset{\mu_{M_q}, \Sigma_{M_q}}{\operatorname{argmax}} \mathcal{L}(\mu_{M_q}, \Sigma_{M_q}; dS^O = \hat{\mu}_{dS^B}) \tag{291}$$

$$\mathcal{L}(\mu_{M_q}, \Sigma_{M_q}; dS^O = \hat{\mu}_{dS^B}) = P_{dS}(\hat{\mu}_{dS^B} | M_q \sim \mathcal{N}(\mu_{M_q}, \Sigma_{M_q})) \tag{292}$$

$$= \prod_m \mathcal{N}(\hat{\mu}_{dS_m^B}; \delta_m \mu_{M_q}, \delta_m(\Sigma_q + \Sigma_{M_q})\delta_m^T)) \tag{293}$$

This result looks quite similar to our original MLE from Section 5.1. The bookmaker implied inconsistency, $\hat{\sigma}^2_{dS_m^B}$, can be utilized to estimate Σ_q, with methods from Appendix D.

The second approach in utilizing dS^B, is by using it as an additional observation in any other estimation method we developed. We can simply input this information into the models described in Sections 5.2 and 5.6, by using the strength difference provided by the bookmaker, dS^B, as an observation of the strength difference. we make a new KPI, *the bookmaker implied strength difference*, and thus define the observations for this KPI as $dS_m^O = \mu_{dS_m^B}$. The main advantage of using bookmaker implied strength differences is that these do not contain any noise. Our current inputs, the number of goals scored in matches, do contain a lot of noise. We see this because outcomes can be very different than the real performance of teams; i.e. due to random factors, a worse performing team can still win. The bookmaker's odds provide us with an outcome distribution rather than a single outcome. Outcomes have a large variance and low bias, while bookmakers predictions have low variance and some bias. As discussed; the bias introduced by bookmakers must be quite low, due to financial incentives of the bookmakers.

5.8 Dynamic player qualities

In the previous sections we developed estimators for static player qualities, but in reality, the qualities of players will change over time. It is possible to assume that player quality will change randomly, while in reality there might be a general relationship between age and player quality. For football we have that, in general, field players achieve their peak quality at the age of 25-27, after this age, their quality tends to decline (Dendir, 2016). For goalkeepers, this peak lies at a later age, due to lower amounts of endured physical stress.

The idea is to design update equations, that apply time-effects between the analysis of two batches. In general, we can say that the mean quality is affected by player effects a, and the player quality uncertainty is affected by passed time and frequency of activity during a period, summarized in the variable τ. We get the following:

$$\hat{\mu}'_{M_q} = \hat{\mu}_{M_q} + \phi_A(a) \tag{294}$$

$$\hat{\Sigma}'_{M_q} = (1 - \phi_T(\tau))\hat{\Sigma}_{M_q} + \phi_T(\tau)\Sigma_q^0 \tag{295}$$

We will choose:

$$\phi_A(a) = 0 \tag{296}$$
$$\phi_T(\tau) = \exp(-\alpha\tau) \tag{297}$$

This way we assume that player qualities evolve non-directional, random, but accumulate variance over time. We can now analyze the performance of our algorithms under these circumstances. Because the mean μ_{M_q} is unaffected, all the unbiasedness properties we have proven will remain the same.

5.9 Discussion of methods

Throughout Section 5 we have developed several estimation methods for the parameters of our model. In this subsection, we will discuss the advantages and disadvantages of the methods, and explain which method we chose to apply to our dataset.

Both the OLS and GLS procedures give unbiased estimates of μ_{M_q} and Σ_{M_q}. They are reasonably simple to implement, and inference can be done in batches. Even on a match-by-match basis, the algorithm would be computationally feasible, by using the algorithm described in Equation (253). The main disadvantage of these methods is the requirement that the performances of players, i.e. the columns of δ, are independent. In general, players quite often perform within a very similar team, only slightly differing columns can lead to extreme overfitting of the parameters to the data. All methods will be influenced by co-linearity, so with a limited dataset we believe some restrictions on the estimated parameters must be applied. Using prior distributions or regularizing the parameters is necessary to avoid overfitting.

Both the Ridge Regression and the conditional Gaussian Inference method deal with this problem by taking a prior distribution over the mean player quality. We chose to implement the conditional Gaussian inference method. The main reason is that this method is the MMSE estimator that is asymptotically unbiased. Also; this method gives us an analytical posterior estimator of the dense matrix Σ_{M_q}. This matrix contains a lot of information about the estimator dependencies in our model, and we have seen that optimizing its elements numerically is very inefficient.

6 Results

In this section we will summarize the achieved results. We focused on developing the Conditional Gaussian Inference method discussed in 5.6. We applied the implementation to our dataset with historical football data. The complete dataset contains 450.000 matches in 580 competitions, for 150.000 players and 13.500.000 player-match objects. After some consideration, we decided to focus on implementing methods that can deal with all the data related to a single competition. In our dataset, most competitions have approximately 3000 unique active players (from 2003-2017), but there are some regional amateur competitions that have much more unique players. We have applied our model to 33 competitions, all of these are listed in Table 6.
The complete results can be downloaded from the following link:
`https://drive.google.com/open?id=1EnEOnhaWBJvEvUjSee4TMPYBvxku1Z1S`

6.1 Model parameters

In this subsection, we will discuss how we determined parameters we used as input for our model.
Firstly we decided to use two individual player qualities; Attack and Defense. These two player qualities are in line with the two main objectives within a football match; scoring goals and not conceding goals.
For an individual match, we have considered two KPI's: goals scored by the home team and goals scored by the away team. These two metrics are the most important for a match; they completely determine the outcome of the match. There is some dependence between home goals and away goals, but nonetheless, we assumed that these measurements are independent.
The most important parameter we choose is the player-quality influences on the match-KPI's; the $\delta_{(\cdot)}$ that form the matrix δ. It must be determined by applying domain knowledge; finding an importance relationship between player roles, player qualities and match-KPI's.
We choose according to Assumption a.8 that $\delta_{(\cdot)} > 0$ if a player's intention is to increase a KPI and $\delta_{(\cdot)} < 0$ if a player's intention is to decrease a KPI. In most cases, a player has no influence on a KPI, because he does not participate in a match, which leads to $\delta_{(\cdot)} = 0$. In our case, home players get positive $\delta_{(\cdot)}$ for home goals, and negative $\delta_{(\cdot)}$ for away goals (opposite for away players). The magnitude of δ is equal to the fraction of minutes played multiplied by the positional factor. These positional factors are different for the KPI's, shown in Table 2. Positional factors are dependent on the match, because a single player can have a different position within different matches. We get:

$$\text{Sign}(i,k,m) = \begin{cases} +1 & \text{if player } i \text{ in match } m \text{ wants to increase KPI } k \\ -1 & \text{if player } i \text{ in match } m \text{ wants to decrease KPI } k \end{cases} \tag{298}$$

$$\text{Minutes\%}(i,m) = \frac{\text{minutes by player } i \text{ in match } m}{\text{total minutes in match } m} \tag{299}$$

$$\text{Positional}(i,j,k,m) = \text{positional factor for quality } j \text{ of player } i \text{ for KPI } k \text{ in match } m \tag{300}$$

$$\delta_{(i,j,k,m)} = \text{Sign}(i,k,m) \cdot \text{Minutes\%}(i,m) \cdot \text{Positional}(i,j,k,m) \tag{301}$$

$$\text{We use: player } i,\ \text{quality } j,\ \text{KPI } k \text{ and match } m$$

The method that we implemented, the Conditional Gaussian Inference method from Section 5.6, requires prior values for player quality mean and player quality variance. It also needs covariance matrix representing the player performance inconsistency. We assumed that prior error in our estimate mean is equal to the performance inconsistency of players; therefore $(\sigma^0_{M_q})^2 = \sigma_q^2$. This gives us that the player inconsistency covariance matrix $\Sigma_q = \sigma_q^2 I$.
In the current runs we have initiated all player quality distributions with $m_q^0 = 0$ and $(\sigma^0_{M_q})^2 = 0.0833$ and $\sigma_q^2 = 0.0833$. These parameters were chosen according Section 4.1. The idea is that we chose an average strength difference mean and variance, these are the population mean and variance. These parameters must also hold for a random match; thus in our initial state. We defined the mean of the strength difference of a random match as $\bar{\mu}_{dS} = 0$ and the variance of the strength difference of a random match as $\bar{\sigma}^2_{dS} = 1$. We have that $E[dS^0] = \bar{\mu}_{dS} = P \cdot m_q^0$ and $var[dS^0] = \bar{\sigma}^2_{dS} = \sqrt{P} \cdot ((\sigma^0_{M_q})^2 + \sigma_q^2)$, where P is equal to the average Euclidian norm of the vector δ_m for all matches. By analysis of the data, we found $P \approx 6$. From this, we yield $m_q^0 = 0$ and $\sigma^0_{M_q} = \sigma_q^2 \approx \frac{1}{12} \approx 0.833$.
Our initial guess for Σ_{M_q} is a diagonal covariance matrix, so all player quality inconsistencies are equal and independent. After processing data the algorithm will find covariances between player quality estimates.

The data is processed in batches, each batch containing all matches of a competition in a certain month. This way we create a constantly evolving player rating from month-to-month throughout the whole history.

Table 2: In this table, we show the definition of the Positional(i, j, k, m) function from Equation (300). It represents the relationship between player position and importance of defense and attack. Attacking quality solely contributes to scoring goals and defensive quality solely contributes to preventing conceding goals.

Position	Attack	Defense
Goalkeeper	0	1
Right\Left Central Defender	0.1	0.9
Right\Left Back	0.15	0.85
Central Defensive Midfielder	0.25	0.75
Right\Left Wing Back	0.25	0.75
Midfielder	0.5	0.5
Offensive Midfielder	0.75	0.25
Forward	0.9	0.1
Right\Left Forward	0.9	0.1

6.1.1 Implementation Specifics

The model has been implemented using the programming language Python3. We used of packages for data management (Pandas, NumPy), Scientific Computing (SciPy, NumPy) and Statistics Related functions (SciPy Statistical).
The computer used for the simulations was a Lenovo Thinkpad with i7 Quadcore CPU, 3.5 Ghz to 4.0 Ghz. A single iteration for one month of mathes takes approximately 5 seconds (depending on amount of players and matches). The computation time of the model for a single competition takes approximately 15 minutes.

6.2 Player ranking

The rankings of players within a competition can be compared. We display the top 20 players for the German Bundesliga, English Premier League and the Primera División in tables 3, 5 and 4 respectively. Note that these results currently were only validated by subjective observations; we see popular, well-known, professional players on the top of our rankings.

Example 2. *Due to the fact that we apply our estimation method in batches, we have the evolution of the estimates throughout time. We show the attacking quality of four players from the Primera División over time in Figure 2.*
The quality estimate of Henry stops after 2010 due to the fact that he transferred to a different competition. Some small movements are due to the fact his estimate was still correlated with the estimate of other players.
Rather than only showing the progress of famous players, we decided to find a player with the highest deviation. We chose the following deviation metric $(q^{max} - q^{min})(q^{max})^2(q^{min})^2$*, this way ensuring that a player has a high maximal rating and low minimum rating. The variables* q^{max} *and* q^{min} *are the maximum and minimum estimate value a player achieved throughout his carreer. The player with the highest deviation turned out to be Álvaro Negredo. As we can see in Figure 2 his quality estimate was high for some years, but after that it declined massively. We looked at some simple individual performance data. As he is a striker we looked at goals and minutes played, to calculate the number of goals scored per 90 minutes played. Note that this information was not used in our estimation procedure. We see that Negredo had the following individual performance:*

total goals	*minutes played*	*goals per 90 minutes*	*club*	*active years*
31	*5690*	*0.49*	*Almeria*	*2007, 2008*
70	*10022*	*0.63*	*Sevilla*	*2009, 2010, 2011, 2012*
10	*2681*	*0.34*	*Valencia*	*2014, 2015*

We see that the individual performance data also implies that the goal-scoring performance of Álvaro Negredo has become worse over time, which supports the fact that our estimate of attacking quality declined.

Table 3: Top 20 players of the German Bundesliga. Only players that were active in 2017 and played at least 2000 minutes in total are shown.

Name	Most played club	Attack	Defense	Weighted Rating
F. Ribéry	FC Bayern München	0.59	0.29	0.46
A. Robben	FC Bayern München	0.69	-0.1	0.36
Y. Poulsen	Rasen Ballsport Leipzig	0.4	0.02	0.34
L. Piszczek	BV Borussia 09 Dortmund	0.34	0.31	0.32
S. Kolašinac	FC Schalke 04	-0.01	0.46	0.29
Luiz Gustavo	VfL Wolfsburg	0.49	0.14	0.28
N. Müller	Hamburger SV	0.31	0.2	0.27
O. Baumann	SC Freiburg	0	0.26	0.26
M. Kruse	Borussia VfL Mönchengladbach	0.31	0.01	0.26
P. Herrmann	Borussia VfL Mönchengladbach	0.28	0.24	0.26
R. Lewandowski	BV Borussia 09 Dortmund	0.3	0.01	0.26
M. Matip	FC Ingolstadt 04	0.07	0.34	0.26
D. Abraham	Eintracht Frankfurt	-0.09	0.4	0.25
G. Castro	TSV Bayer 04 Leverkusen	0.7	-0.22	0.25
S. Rudy	TSG 1899 Hoffenheim	0.31	0.21	0.25
M. Compper	TSG 1899 Hoffenheim	0.27	0.24	0.25
R. Bürki	BV Borussia 09 Dortmund	0	0.24	0.24
Javi Martínez	FC Bayern München	0.11	0.31	0.24
E. Durm	BV Borussia 09 Dortmund	-0.09	0.41	0.24
F. Sørensen	1. FC Köln	0.08	0.3	0.24

Table 4: Top 20 players of the Spanish Primera División. Only players that were active in 2017 and played at least 2000 minutes in total are shown.

Name	Most played club	Attack	Defense	Weighted Rating
L. Messi	FC Barcelona	0.6	-0.05	0.5
Bruno González	Real Betis Balompié	0.15	0.49	0.38
Cristiano Ronaldo	Real Madrid Club de Fútbol	0.39	0.1	0.34
M. Krohn-Dehli	Real Club Celta de Vigo	0.32	0.34	0.33
Lucas Vázquez	Real Madrid Club de Fútbol	0.39	0.18	0.32
Juanfran	Club Atlético de Madrid	-0.13	0.47	0.3
G. Bale	Real Madrid Club de Fútbol	0.34	0.07	0.29
Aleix Vidal	UD Almería	0.33	0.2	0.28
M. ter Stegen	FC Barcelona	0	0.27	0.27
Alejandro Gálvez	Rayo Vallecano	0.08	0.35	0.27
Marco Asensio	Reial Club Deportiu Espanyol	0.39	-0.08	0.26
Dani Parejo	Valencia Club de Fútbol	0.47	0.05	0.25
Bruno	Villarreal Club de Fútbol	0.25	0.24	0.25
Albentosa	Málaga Club de Fútbol	0.04	0.32	0.24
L. Suárez	FC Barcelona	0.29	-0.06	0.24
Soldado	Valencia Club de Fútbol	0.29	-0.05	0.24
Casemiro	Real Madrid Club de Fútbol	0.18	0.27	0.23
Mikel Rico	Athletic Club Bilbao	0.09	0.33	0.23
Pablo Sarabia	Getafe Club de Fútbol	0.15	0.34	0.23
R. Varane	Real Madrid Club de Fútbol	0.17	0.25	0.23

Table 5: Top 20 players of the Premier League. Only players that were active in 2017 and played at least 2000 minutes in total are shown.

Name	Most played club	Attack	Defense	Weighted Rating
M. Dembélé	Tottenham Hotspur FC	0.55	0.42	0.48
A. Valencia	Manchester United FC	0.35	0.55	0.45
M. Darmian	Manchester United FC	-0.02	0.63	0.43
H. Kane	Tottenham Hotspur FC	0.51	-0.02	0.43
J. Henderson	Liverpool FC	0.45	0.33	0.39
P. Cech	Chelsea FC	0	0.39	0.39
D. Sturridge	Liverpool FC	0.45	0.05	0.39
D. Sakho	West Ham United FC	0.41	-0.05	0.33
J. Cork	Swansea City AFC	0.36	0.3	0.32
V. Kompany	Manchester City FC	0.27	0.33	0.31
F. Forster	Southampton FC	0	0.3	0.3
N. Matic	Chelsea FC	0.01	0.45	0.3
David Silva	Manchester City FC	0.33	0.24	0.29
Azpilicueta	Chelsea FC	0.39	0.24	0.29
T. Alderweireld	Tottenham Hotspur FC	0.22	0.31	0.28
M. Özil	Arsenal FC	0.36	0.09	0.28
S. Coleman	Everton FC	0.2	0.31	0.28
Fernandinho	Manchester City FC	0.52	0.12	0.27
Y. Touré	Manchester City FC	0.16	0.35	0.26
J. Lescott	Everton FC	0.56	0.14	0.26

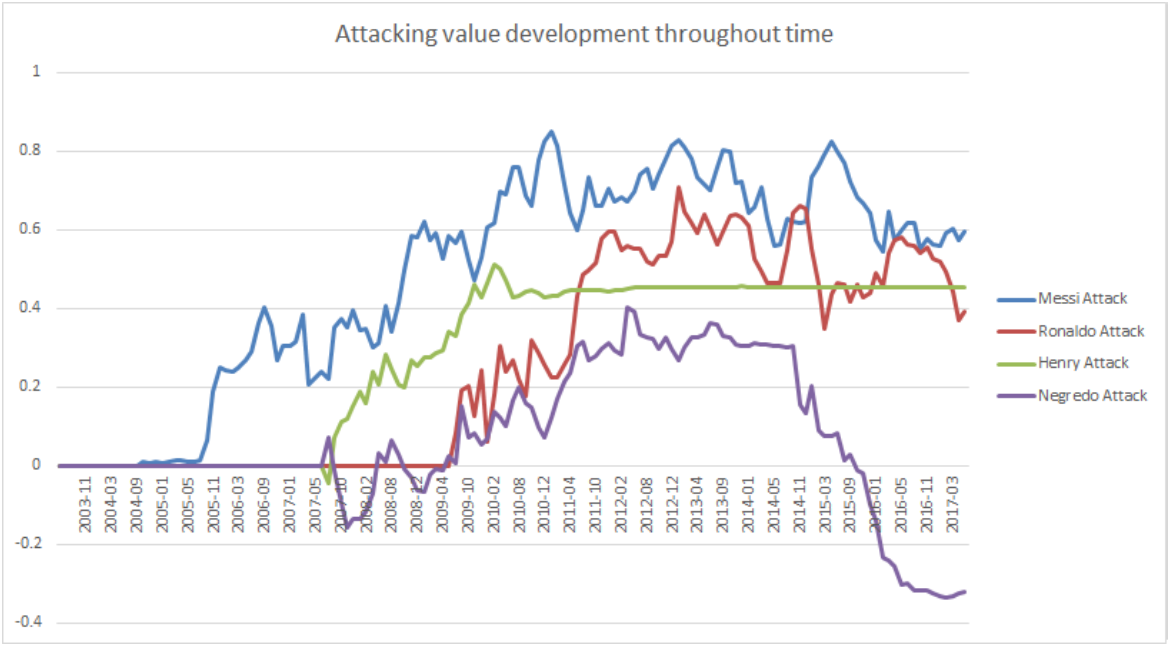

Figure 2: Estimates of the mean attacking quality of Lionel Messi, Cristiano Ronaldo, Thierry Henry and Álvaro Negredo throughout time

6.3 Match outcome prediction

Our model produces outcome probabilities for KPI's, also prior to matches. In our implementation specifically, we yield outcome probability distributions for goals scored by the home and away team. We can compare our predictions with the odds provided by bookmakers. There are several betting markets we could benchmark our predictions with (total goals, total goals over/under, exact score, handicap, Asian handicap), but we chose to simply focus on the match outcomes. A match outcome is either a home win, a draw or away win.

In some research papers, we see the percentage of correctly predicted outcomes as the evaluation criterion. In our opinion this is an incorrect metric to use in this setting where all events are completely different (not identically distributed). For example, whenever a professional team is playing an amateur team, predicting a win for the professional team will achieve an accurate prediction almost always.

A way to evaluate our predictions is by calculating the logarithmic loss (LogLoss). This is a measure of the likelihood of our predictions. The metric logarithmic loss is defined as:

$$LogLoss(\hat{\theta}) = -\frac{1}{N_M}\sum_{m=1}^{N_M} \ln\left(P_{\hat{\theta}}(D_m = d_m)\right) \tag{302}$$

Here we hcae that $P_{\hat{\theta}}(D_m = d_m)$ is our predicted probability for the eventual outcome d_m of match m. The LogLoss of our predictions is shown in Table 6. We expect a dummy model to get a logarithmic loss of 1.05 per match. In some competitions, our model performs quite good, especially in the last year where the parameter estimates are closer to convergence. Unfortunately, we see bad results for some competitions. This is because in some cases our model makes very extreme predictions that turn out to be incorrect.

We believe that the the most challenging benchmark for any prediction algorithm is the existing market opinion. We can find the market opinion, which is publicly available, in the odds provided by bookmakers in the gambling market. We used our predictions to generate a historical betting strategy and test whether the betting strategy would have yielded a profit. We explain the betting strategy in words:

$$\text{Bet 1 unit if Odds} \cdot \text{Probability} > 1.1 \text{ and Probability} > 0.3 \tag{303}$$

The decision to only bet on events with higher than 30% probability was made due to experience with earlier prediction models. Firstly; we believe bookmakers are very difficult to outperform on

low-probability events. Another reason is that betting on low probability events adds a lot of variance to the realized profit. By betting events with a probability $> 30\%$ we keep the variance of the realized profit relatively small, which makes it more reliable.
The bookmaker odds were taken from a purchased dataset, which is collected and distributed by (Indatabet, 2017). The results are also shown in Table 6.
We used historical odds provided by the bookmaker Pinnacle. This bookmaker tends to apply a 2% margin on all their odds. Effectively this means that any random betting strategy is expected to lose 2% of the total amount bet, in the long run.

Table 6: In this table, we present two metrics that measure the quality of our predictions. The model was applied separately for multiple competitions, yielding the average logarithmic loss and the profitability for each. For the logarithmic loss lower values are considered better. The profit is the amount of money won by by wagering one unit per match according to our betting strategy described in (303). The *total bet* column shows how many units were wagered in total. The columns *recent profit* and *recent total bet* refer to the total profit and total amount wagered during the most recent year of betting.

League Name	Country	Profit	Total bet	Recent profit	Recent bet	LogLoss	Recent LogLoss
Eredivisie	Netherlands	-12.31	627	3.25	157	0.98	0.84
Eerste Divisie	Netherlands	43.12	841	21.51	205	1.03	1.27
La Liga	Spain	-39.76	734	-10.13	162	1	0.84
Premier League	England	-40.38	843	-12.12	213	1	0.82
Bundesliga	Germany	32.86	781	6.52	170	1.03	1.14
2. Bundesliga	Germany	9.21	900	1.91	241	1.07	1.04
Segunda División	Spain	24.68	1294	0.72	280	1.07	1.1
Serie A	Italy	17.48	856	-0.02	182	1.01	0.99
Serie B	Italy	-39.64	1340	9.96	325	1.08	1.08
League One	England	67.06	1732	7.5	415	1.06	1.26
Ligue 1	France	-49.87	925	-6.11	221	1.05	0.99
Ligue 2	France	-26.75	956	-39.23	252	1.09	1.04
Süper Lig	Turkey	12.82	794	31.38	166	1.04	0.97
Veikkausliiga	Finland	23.06	437	13.89	81	1.05	0.91
First Division A	Belgium	3.62	712	7.76	166	1.02	0.85
Serie A	Brazil	7.66	677	-2.56	110	1.06	2.78
Super League	Switzerland	27.43	497	-9.01	104	1.03	1.22
Allsvenskan	Sweden	20.5	469	2.82	101	1.03	0.97
Eliteserien	Norway	-2.12	503	8.28	104	1.02	1.13
MLS	USA	17.02	677	27.07	133	1.06	1.04
Premier Division	Republic of Ireland	28.24	329	4.77	80	0.99	1
1. Division	Norway	-10.62	463	-9.91	118	1.04	1.12
Superettan	Sweden	-19.37	518	-11.89	126	1.05	1.13
Premiership	Scotland	10.32	560	24.02	117	1.02	0.83
Championship	Scotland	11.35	409	9.78	87	1.03	1.18
Bundesliga	Austria	12.34	392	7.8	72	1.02	0.92
1. Liga	Austria	-14.26	477	6.54	96	1.04	1.05
CSL	China PR	-39.56	251	-3.5	85	1.05	1.02
First Division B	Belgium	18.68	664	-9.49	111	1.05	1.11
National 1	France	62.25	1003	30.15	242	1.06	0.92
1. HNL	Croatia	-36.08	289	7.35	41	0.98	0.85
Primeira Liga	Portugal	-62.17	590	4.06	178	0.98	1.12
Total		56.81	22540	123.07	5141	1.03	1.08

7 Conclusions and recommendations

In this section, will discuss the results, the applicability and the shortcomings of our model. We will also provide recommendations for further work.
In this thesis, we developed a model for inferring individual player qualities based on group comparisons. We implemented the model for football, calculating the attack and defense qualities for all players based on historical match outcomes. The only information used from each match were line-ups (player positions and playing time), home goals and away goals. The results of our model can be assessed by looking at player rankings and predictions of future match outcomes. The player ranking is a subjective criterion. Professional football scouts can disagree on the quality of a player, even if presented with the same information. Our player ranking results in high qualities and low uncertainty for a lot of well-known, highly skilled players. An objective way to assess the model quality is by looking at the generated predictions for future matches. The betting results show a small positive return, which is much better than the expected -2% loss per bet for a random strategy. The model we developed can be applied in any context where multiple players collaborate and/or compete to influence (increase/decrease) certain match outcomes (Key Performance Indicators, KPI's). Good examples are team-based projects (film-making, crime investigation, project development), online games (Dota, Overwatch, League of Legends, Call of Duty, etc.) and sports (Football, Basketball, Volleyball, Hockey, etc.).
We modeled the relationship between player participation and KPI outcomes. We wanted to use a non-parametric and non-linear approach. For this we created the variable strength difference, a weighted linear combination of player qualities that has a non-linear relationship with the KPI outcome distribution. The weights in this linear combination correspond to the importance of certain player qualities. A weight is positive whenever a player wants to increase a KPI, a weight is negative if a player wants to decrease a KPI and a weight is zero if a player does not have any influence on a KPI. An important assumption is that we determine these weights deterministically by using domain knowledge. In the application to football we used a combination of domain knowledge and participation data to determine the weights. The non-linear relationship between the strength difference and KPI outcomes is modeled by an IRT model, with some parameters that are estimated from the data.

7.1 Assumptions and shortcomings

We assumed that player qualities are additive and independent. Firstly; this means we assume there is no auto-correlation between performance, while a lot of professionals believe there is some. Individuals are referred to having a good/bad streak and being in/out of form. In basketball, this effect, within a match for a certain player, is called "the hot-hand". Multiple research suggests that this effect does not exist, therefore it's often referred to as "the hot-hand fallacy" (Gilovich *et al.* , 1985).
We assumed that player qualities are additive and inconsistencies are independent, effectively we do not account for multi-player synergistic effects. We explicitly chose not to model such effects; the main reason being that there are too many multi-player coalitions to consider. Within a team of N players there are $\frac{N(N-1)}{2}$ pairs of players. It is infeasible to model so many effects properly; for a football match within a single team there are 66 duos and 220 triplets of players within one team.
Even though we used a large dataset, we believe more data would yield better results. Ideally, we would have a very diverse and rich data; huge amounts of comparisons with constantly changing participation structures. This would give us the ability to find all relationships between the quality of players, and potentially even identify synergistic effects. In the current dataset we observe that quite often players tend to perform within a very similar coalition. In football, players are affiliated with a single team, and therefore they will only play together with other players affiliated to this team against opponents from a single competition. This can lead to co-linearity, and thus indistinguishability between players. This is an important reason why we implemented a method with prior distributions over unknowns in our model; to avoid overfitting player quality parameters to the data.
Throughout the research, we explored the idea of improving our estimates by using subjective observations; the implied strength differences based on bookmaker odds. The bookmaker's odds are a result calculated by the global betting market, we argue that they are correct due to the efficient market hypothesis. Using the information provided by bookmakers can help dealing with unknown or rapidly changing player qualities; bookmakers incorporate additional information rather than only historical results. We did not apply this idea to the results presented in this research.
A large problem, that we did not solve in this research is the estimation of competition quality. Teams within a competition have several encounters, while teams from different competitions almost never

encounter one another. We applied our model to individual competitions separately, disregarding any matches played in a different competition. This means an experienced player, that goes to a new competition will be treated as an unknown/random player in our current model. We chose to do this due to computational limitations for the methods when applied to multiple competitions. Cross-competitional quality estimation is very difficult and should be an important focus for further research. Having accurate competitional qualities can lead to the efficient application of our model over multiple competitions and overall improvements of the results due to a richer performance history for all players.

7.2 Recommendations for future work

The current model was formulated abstractly to be applicable in all situations where individuals participate, but only team results are observed. This corresponds to the type of data collected in the film-making business; the credits describe who participated in film and the overall success of a film could be measured, but individual contributions are not recorded in general. We will apply the model in the near future to calculate the qualities of professionals in the film industry (cast, director, producer) by using the IMDb dataset.
Within this project we extensively discussed the possibility to utilize bookmaker predictions as an input to the model. We see this type input as very useful, due to the fact that it contains almost no variance. A next step can be to add the information implied from bookmaker odds to the model, hopefully yielding even better results.
The project was in collaboration with SciSports, therefore one of the desired goals was to create a model that can be applied industrially by SciSports within the world of football analytics. Currently, SciSports already has an individual player quality model, the SciSkill, producing player quality estimates for 70.000 active players in dozens of leagues. The SciSkill algorithm is a difference-based approach, inspired by the ELO model, and it produces other player metrics like potential rating and resistance (experience) factor. The next version of the SciSkill model, due summer 2018, will contain insights, ideas, and algorithms that we developed throughout this research. Within this research we have focused on analytical mathematical techniques, during implementation it should be considered to use certain non-analytic techniques like Gibbs-sampling, machine learning algorithms or methods relying on numerical integration to calculate the player quality estimates.

8 Table with variable definitions

Table 7: Table with variable definitions

Variable	Explanation
$Q_{(i,j)}$	Random variable representing the quality j of player i
$Q_{(i)}$	Column vector containing all $Q_{(i,j)}$ for player i
Q	Column vector containing $Q_{(i,j)}$ for all qualities of all players
$M_{(i,j)}$	Random variable for the mean of quality j of player i
$M_{(i)}$	Column vector containing all $M_{(i,j)}$ for player i
M_q	Column vector containing all $M_{(i,j)}$ for all players
$\sigma^2_{(i,j)}$	Variance of $Q_{(i,j)}$
Σ_{q_i}	Covariance of column vector $Q_{(i)}$
Σ_q	Covariance of column vector Q
μ_{M_q}	Mean of column vector M_q
Σ_{M_q}	Covariance of column vector M_q
$\delta_{(i,j,k,m)}$	Weight of quality j of player i in match m for KPI j
δ_m	Row vector containing all $\delta_{(\cdot)}$ related to match m
δ	Matrix containing all the $\delta_{(i,j,k)}$
dS	Vector containing the strength difference for all matches, often represented as δQ
dS_m	Strength difference in a match m, often represented as $\delta_m Q$
$D_{k,m}$	Ordinal random variable of the outcome of KPI k in match m
D_m	Column vector containing all $D_{k,m}$ for match m
D	Column vector containing all $D_{k,m}$ for all matches
ξ_k	General mapping from strength difference to outcome probability of KPI k
β	Benchmark parameters that are required for the function ξ
$d_{k,m}$	Observation of random variable $D_{k,m}$
d	Observation of random variable D
dS^O_m	Observation of strength difference in a match m; follows from the observation d of D
$u(\cdot)$	Utility function, required for the function ξ
$E[X]$	Expected value of random variable X
$Var[X]$	Variance of random variable X
$\mathcal{N}(\mu, \sigma^2)$	Gaussian random variable with mean μ and variance σ^2
$\mathcal{N}(x; \mu, \sigma^2)$	Probability density function of Gaussian random variable with mean μ and variance σ^2
$\mathcal{U}[a, b](x)$	Probability density function of uniform random variable
$P_\theta(D = d)$	Probability distribution of D, whenever we choose θ as the parameters of our model
$p_X(x)$	Probability density function of random variable X
$\mathcal{L}(\theta = \hat{\theta}; D = d)$	Likelihood of parameter estimate $\hat{\theta}$, as a function of observed outcome d of random variable D
$\mathbb{D}_{KL}(P\|Q)$	KL divergence of probability distribution P with respect to probability distribution Q
dS^B_m	Strength difference implied by bookmaker odds
N_P, N_M, N_K, N_Q	Total number of players, matches, KPI's per match and qualities per player, respectively
D	Vector of random variables that represent the KPI outcomes
$D_{k,m}$	Random variable that represents the outcome of KPI k in match m
$\mathbb{R}$	Real Numbers
$\mathbb{D}$	Ordinal outcome space of all the KPI's in a match
$\mathbb{D}_k$	Ordinal outcome space of KPI type k
Y_{ij}	Outcome of match for i against j
$R_i^{(m)}$	DSS game outcome rating (KNTB Tennis)
θ	Variable that contains all parameters of our model that need to be estimated
$\hat{\theta}$	Estimator of θ
ϵ	Estimation error; the difference between estimated and realized outcome
$\phi_A(a)$	function that affects our estimate of the player quality mean as a function of player effect a
$\phi_T(\tau)$	function that affects the estimate of the player quality covariance as a function of time effects τ
I	Identity Matrix
$1(X)$	Indicator function, equal to 1 if X is true
$1_{[a,b]}(x)$	Indicator function, equal to 1 if $x \in [a, b]$
$\vec{\mathbb{1}}$	Column vector filled with ones
$A \circ B$	Elementwise multiplication of matrices A and B

A KPI outcome probability model

In this section, we will further clarify our approach to determining the probability distribution of a KPI outcome. Firstly we will explain the intuition behind our complete approach, then we will visualize the procedure with some numerical examples and finally, we will apply the procedure to a Poisson distribution where the mean is a latent variable.

A.1 Intuitive explanation of methodology

The goal of this methodology to determine a mapping from participants in a match to outcome probabilities. Firstly we have created latent variables for all players that numerically describe their power in a certain quality. We assume that the utility, or strength, of a team, can be seen as a weighted sum of individual qualities. This difference between two coalitions in a certain match is the strength difference, in short notation for match m we have $dS_m = \delta_m Q$.
There is a general, intuitive relationship between outcome probabilities and strength difference; stronger teams perform better and therefore have an outcome probability with higher (lower) mean for positive (negative) events. An event is considered positive if a team wants it to happen. We have by assumption that strength has a normally distributed around its average value, $dS \sim \mathcal{N}(\mu_{dS}, \sigma^2_{dS})$.
The method in Section 4 was constructed with the following idea;

$$P(D = k|dS = s) = \begin{cases} 1 & \text{if } \ s \in [\beta_{k-1}, \beta_k) \\ 0 & \text{else} \end{cases} \tag{304}$$

Intuitively we can now see that if the link function gives a correct representation of the strength difference outcomes, we can keep the thresholds the same for all different, match dependent, strength difference distributions.

A.1.1 Visualisation of Gaussian distribution as a link function

To clarify the approximation procedure from Section 4.2.4 we made two pictures that illustrate a simple example. In our example, we look at 3 ordered categories; category 1, category 2 and category 3. We choose average class probabilities 50%, 26%, and 24% respectively. In figure 3 we show the extracted of β_k given class probabilities P_{C_k}, as in equation (97). We get $\beta_1 = 0$ and $\beta_2 = 0.7$, while $\beta_0 = -\infty$ and $\beta_3 = \infty$ are implied.
Now we have a new observation and we want to determine the category probabilities for it. We take that for this record we have a strength discrepancy of $w^T q = 0.5$. We yield figure 4, and can extract the class probabilities, while leaving the β_i's unchanged. We get $\bar{P}_{C_1} = 0.31, \bar{P}_{C_2} = 0.27$ and $\bar{P}_{C_k} = 0.42$.

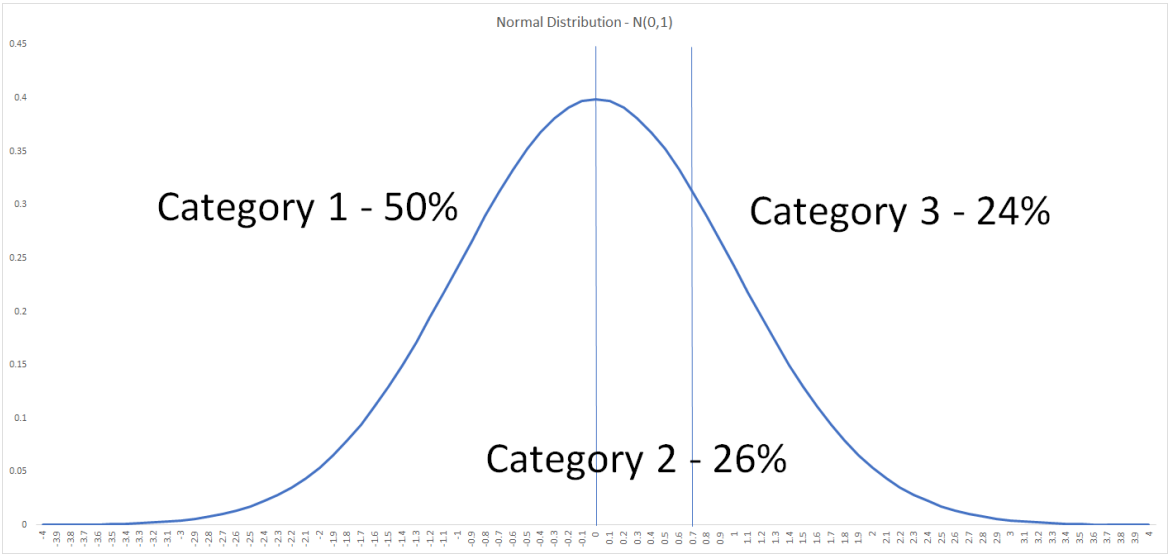

Figure 3: Standard class distribution

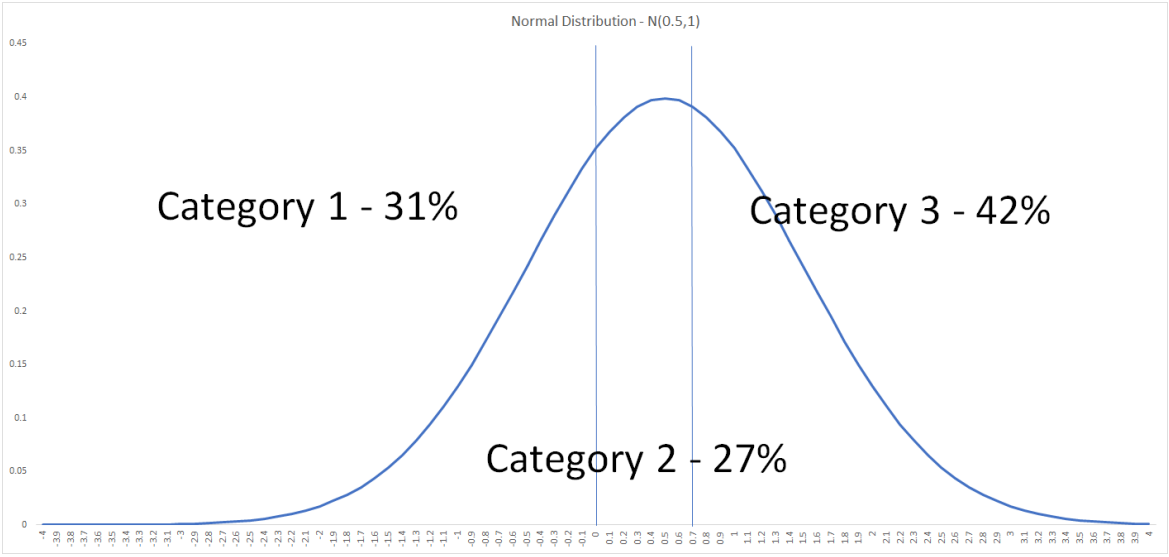

Figure 4: Class distribution for specimen with $\delta M_q = 0.5$

A.1.2 Applicability to Poisson distribution

We will now apply the procedure the procedure, discussed in Appendix A.1 and developed in Section 4, to Poisson Distribution with *a variable mean*. The idea is that the mean is a function of a latent variable. We will show that the approximate probability distribution we yield with our method is similar to the real probability distribution. We compare our approximation and the real distributions by looking at two metrics: the Kullback-Leibler Divergence (Kullback & Leibler, 1951) and Total Variation. We summarize the procedure as follows:

1. First, we choose an appropriate mean for our central distribution. Whenever $\delta M_q = 0$, our approximated distribution will be exact.
2. Now we extract the β's by using equation (97)
3. We calculate the class probabilities $P_{C_k}(x) = P(D = k|\delta_m M_q = x)$ according to equation (98), leaving δM_q as a parameter
4. With the calculated $P_{C_k}(x)$ we extract the mean of our probability distribution, $\bar{\mu}(x) = \sum_k k P_{C_k}(x)$, numerically
5. Now we calculate the probabilities should expect, $P^R_{C_k}(x)$ given the variable has a Poisson distribution, by taking
$$P^R_{C_k}(x) = P(Y = k|Y \sim Poiss(\bar{\mu}(x))). \tag{305}$$
6. We calculate the difference between the real and estimated probability distribution with metrics like Total Variation and the KL-divergence

Even though the probability distributions are in-exact, we yield decent approximations the Poisson distribution. If we take $\mu_0 = 1.7$ and $\delta M_q = x = 1$, this implies $\bar{\mu}(1) = 3.12$, and yields a total variation (TV) of 0.089. This is calculated by:

$$\text{Total Variation} = TV = \sum_k |P^R_{C_k}(x) - \bar{P}_{C_k}(x)| \tag{306}$$

The Kullback-Leibler Divergence is calculated by the following equation:

$$\mathbb{D}(P^R||\bar{P})(x) = \sum_k P_{C_k}(x) \log\left(\frac{P_{C_k}(x)}{\bar{P}_{C_k}(x)}\right) \tag{307}$$

The results are summarized in Table 8 and 9.

Table 8: Distribution approximation performance for $\delta M_q = 1$, implying $\mu = 3.12$

k	P_{C_k}	$\sum_{i=1}^{k} P_{C_i}$	β_{k+1}	$\bar{P}_{C_k}$	$Poiss(\bar{\mu})$	$\lvert P^R_{C_k} - \bar{P}_{C_k}\rvert$	KL Divergence
0	18.27%	18.27%	-0.91	2.84%	4.42%	1.59%	0.85%
1	31.06%	49.32%	-0.02	12.62%	13.79%	1.17%	0.53%
2	26.40%	75.72%	0.70	22.65%	21.51%	1.14%	-0.48%
3	14.96%	90.68%	1.32	24.49%	22.35%	2.14%	-0.89%
4	6.36%	97.04%	1.89	18.63%	17.43%	1.20%	-0.51%
5	2.16%	99.20%	2.41	10.83%	10.87%	0.04%	0.02%
6	0.61%	99.81%	2.90	5.06%	5.65%	0.59%	0.27%
7	0.15%	99.96%	3.36	1.97%	2.52%	0.54%	0.27%
8	0.03%	99.99%	3.80	0.66%	0.98%	0.32%	0.17%
9	0.01%	100.00%	4.22	0.19%	0.34%	0.15%	0.09%
10	0.00%	100.00%	4.63	0.05%	0.11%	0.06%	0.04%
11	0.00%	100.00%	5.02	0.01%	0.03%	0.02%	0.01%

Table 9: TV and KL divergence of the approximation of the Poisson distribution for different values of δM_q

δM_q	$\bar{\mu}$	Total Variation	KL divergence
-2	0.16	5.3%	0.30%
-1.5	0.36	9.0%	0.39%
-1	0.68	9.1%	0.29%
-0.5	1.13	6.0%	0.10%
0	1.70	0.0%	0.00%
0.5	2.37	4.5%	0.10%
1	3.12	8.9%	0.37%
1.5	3.95	11.5%	0.75%
2	4.84	13.2%	1.17%
2.5	5.81	14.5%	1.62%
3	6.84	15.4%	2.07%
3.5	7.93	16.1%	2.51%
4	8.15	15.8%	2.60%

A.2 Bayesian two-dimensional rating inference

Consider two players, player 1 and player 2, take respective ratings $\theta_1 \sim \Theta_1$ and $\theta_2 \sim \Theta_2$. For this specific case Bayes' rule resorts to:

$$P(\Theta_1 = \theta_1, \Theta_1 = \theta_2 | Y_{1,2} = y) = \frac{P(Y_{1,2} = y, \Theta_1 = \theta_1, \Theta_2 = \theta_2)}{P(Y_{1,2} = y)} \tag{308}$$

$$= \frac{P(Y_{1,2} = y | \Theta_1 = \theta_1, \Theta_2 = \theta_2) P(\Theta_1 = \theta_1, \Theta_2 = \theta_2)}{\int\limits_{x_1} \int\limits_{x_2} P(Y_{1,2} = y | \Theta_1 = x_1, \Theta_2 = x_2) P(\Theta_1 = x_1, \Theta_2 = x_1) dx_1 dx_2} \tag{309}$$

We assume that player ratings behave over time according to a Brownian motion. This means that ratings have a normal distribution, with an increasing variance during periods of inactivity. Whenever there is a result, we gain information and can, therefore, update our estimator of the current location of *the Brownian Motion*.
This gives us that $\theta_1 \sim \Theta_1 = \mathcal{N}(\mu_1, \sigma_1^2)$ and $\theta_2 \sim \Theta_2 = \mathcal{N}(\mu_2, \sigma_2^2)$. We assume a logistic outcome probability distribution:

$$P(Y_{1,2} = 1 | \Theta_1 = \theta_1, \Theta_2 = \theta_2) = \frac{1}{1 + e^{\theta_2 - \theta_1}} \tag{310}$$

Equation (310) represents the relationship between outcome probability and player ratings. We discuss other and generalized choices of such relationships in Section 4.1.

Combining Equations (309) and (310) gives us the following relationship:

$$P(\Theta_1 = \theta_1, \Theta_2 = \theta_2 | Y_{1,2} = y) = \frac{\frac{1}{1+e^{\theta_2-\theta_1}}\mathcal{N}(\theta_1; \mu_1, \sigma_1^2)\mathcal{N}(\theta_2; \mu_2, \sigma_2^2)}{\int\limits_{x_1}\int\limits_{x_2} \frac{1}{1+e^{x_2-x_1}}\mathcal{N}(x_1; \mu_1, \sigma_1^2)\mathcal{N}(x_2; \mu_2, \sigma_2^2)dx_1 dx_2} \tag{311}$$

$$= \frac{\frac{1}{1+e^{\theta_2-\theta_1}}\mathcal{N}(\theta_1; \mu_1, \sigma_1^2)\mathcal{N}(\theta_2; \mu_2, \sigma_2^2)}{\int\limits_{D} \frac{1}{1+e^{-D}}\mathcal{N}(D; \mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)dD} \tag{312}$$

$$\approx \frac{e^{-g(\sigma_1,\sigma_2)(\theta_1-\theta_2)}}{1+e^{g(\sigma_1,\sigma_2)(\theta_2-\theta_1)}}\mathcal{N}(\theta_1; \mu_1, \sigma_1^2)\mathcal{N}(\theta_2; \mu_2, \sigma_2^2) \tag{313}$$

In Equation (313) we have that $g(\sigma_1, \sigma_2) = \sqrt{1 + \frac{\pi(\sigma_1^2+\sigma_2^2)}{8}}$, which is the result of procedure to approximate a logistic integral (Crooks, 2013), that has also been applied in the Glicko-model (Glickman, 1999). The multiplication of Gaussian pdfs is discussed in Appendix C.2.

B Estimators

In this section we will prove, or refer to proofs of statements made throughout the report regarding estimators.

B.1 Bias-variance decomposition

The bias and variance of an estimator are very important properties. We will define them, and prove the bias-variance decomposition of the MSE of an estimator.

$$\text{Var}(\hat{\theta}) = E\left[\left(\hat{\theta} - E[\hat{\theta}]\right)^2\right] \tag{314}$$

$$\text{bias}(\hat{\theta}, \theta) = E\left[\hat{\theta} - \theta\right] = E[\hat{\theta}] - \theta \tag{315}$$

$$\text{MSE}(\hat{\theta}) = E\left[(\hat{\theta} - \theta)^2\right] \tag{316}$$

$$= E\left[\left(\hat{\theta} - E[\hat{\theta}] + E[\hat{\theta}] - \theta\right)^2\right] \tag{317}$$

$$= E\left[\left(\hat{\theta} - E[\hat{\theta}]\right)^2\right] + E\left[2\left(\hat{\theta} - E[\hat{\theta}]\right)\left(E[\hat{\theta}] - \theta\right)\right] + E\left[\left(E[\hat{\theta}] - \theta\right)^2\right] \tag{318}$$

$$= E\left[\left(\hat{\theta} - E[\hat{\theta}]\right)^2\right] + 2\left(E[\hat{\theta}] - E[\hat{\theta}]\right)\left(E[\hat{\theta}] - \theta\right) + \left(E[\hat{\theta}] - \theta\right)^2 \tag{319}$$

$$= \text{Var}(\hat{\theta}) + \text{bias}(\hat{\theta}, \theta)^2 \tag{320}$$

B.2 Minimum mean squared error

Here we will prove uniqueness of the minimum MSE estimator and show that $\hat{\theta}^{MMSE} = E[\theta|\text{Data}]$. We will denote our data by Y and our estimator by $g(Y) = \hat{\theta}$.

$$\text{MSE} = E\left[\left(\theta - \hat{\theta}\right)^2 \middle| Y = y\right] = E\left[(\theta - g(Y))^2 \middle| Y = y\right] \tag{321}$$

$$= E\left[\theta^2 \middle| Y = y\right] - 2E\left[\theta g(Y) | Y = y\right] + E\left[g(Y)^2 | Y = y\right] \tag{322}$$

$$= E\left[\theta^2 \middle| Y = y\right] - 2g(y)E\left[\theta | Y = y\right] + g(y)^2 \tag{323}$$

$$\frac{\text{dMSE}}{\text{d}g(y)} = -2E\left[\theta | Y = y\right] + 2g(y) = 0 \tag{324}$$

$$g(y) = E\left[\theta | Y = y\right] \tag{325}$$

C Gaussian random variables

In this section of the appendix we will elaborate certain analytical and approximation methods we have applied throughout this research regarding Gaussian random variables.

C.1 Truncated Gaussian distribution

In some cases we can deal with a normally distributed random variable, for which we know it must lie in a (bounded or unbounded) range. The probability distribution of such a random variable is characterized by the truncated normal distribution. The parameters of a truncated normal distribution were taken from (Burkardt, 2004) and are listed here:

$$Z \sim \mathcal{N}(\mu, \sigma^2) \tag{326}$$

$$X \sim Z|Z \in [a, b] \tag{327}$$

$$f(x; \mu, \sigma, a, b) = \frac{\phi(\frac{x-\mu}{\sigma})}{\sigma\left(\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})\right)} \tag{328}$$

$$E[X] = \mu + \frac{\phi(\frac{a-\mu}{\sigma}) - \phi(\frac{b-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}\sigma \tag{329}$$

$$Var(X) = \sigma^2\left[1 + \frac{\frac{a-\mu}{\sigma}\phi(\frac{a-\mu}{\sigma}) - \frac{b-\mu}{\sigma}\phi(\frac{b-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} - \left(\frac{\phi(\frac{a-\mu}{\sigma}) - \phi(\frac{b-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}\right)^2\right] \tag{330}$$

C.2 Multiplication of Gaussian PDFs

In a few places throughout our research we need to find the product of (multivariate) Gaussian probability density functions. We know that if $X \sim \mathcal{N}(\mu, \Sigma)$, then:

$$p_X(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \tag{331}$$

By using (Bromley, 2014), we get the following results:

$$\prod_{i=1}^{n}\mathcal{N}(\mu_i, \Sigma_i) = \exp\left(\left(\sum_{i=1}^{n}\zeta_i\right) - \zeta_n\right)\exp\left(\zeta_n + (\sum_{i=1}^{n}\Sigma_i\mu_i)^T x - \frac{1}{2}x^T(\sum_{i=1}^{n}\Sigma_n)x\right) \tag{332}$$

$$\zeta_i = -\frac{1}{2}\left(d\log 2\pi - \log|\Sigma_i| + (\Sigma_i\mu_i)^T\Sigma_i^{-1}(\Sigma_i\mu_i)\right) \tag{333}$$

For the special case of $n = 2$ we get:

$$\mathcal{N}(x; \mu_1, \Sigma_1)\mathcal{N}(x; \mu_2, \Sigma_2) = \mathcal{N}(\mu_1; \mu_2, \Sigma_1 + \Sigma_2)\mathcal{N}(x; C(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2), C) \tag{334}$$

$$C = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \tag{335}$$

Note that the integrate over the multiplication of two Gaussians pdf, we yield the following:

$$\int_{-\infty}^{\infty}\mathcal{N}(x; \mu_1, \Sigma_1)\mathcal{N}(x; \mu_2, \Sigma_2)dx = \mathcal{N}(\mu_1; \mu_2, \Sigma_1 + \Sigma_2) = \mathcal{N}(\mu_2; \mu_1, \Sigma_1 + \Sigma_2) \tag{336}$$

C.3 Multivariate Gaussian distribution with multivariate Gaussian mean

Imagine that we have:

$$p_X(x) = \mathcal{N}(x; \mu_x, \Sigma_x) \tag{337}$$

$$p_{Y|X}(y) = \mathcal{N}(y; AX + b, \Sigma_y) \tag{338}$$

Here we assume that A is invertible, and we yield the following result:

$$p_Y(y) = \int_{-\infty}^{\infty} f_{Y|X}(y|x) f_X(x) dx \tag{339}$$

$$= \int_{-\infty}^{\infty} \mathcal{N}(y; Ax + b, \Sigma_y) \mathcal{N}(x; \mu_x, \Sigma_x) dx \tag{340}$$

$$= \int_{-\infty}^{\infty} C \cdot exp(-\frac{1}{2}((Ax + b - y)^T \Sigma_y^{-1} (Ax + b - y) + (x - \mu_x)^T \Sigma_x^{-1} (x - \mu_x)) \tag{341}$$

$$= \int_{-\infty}^{\infty} C \cdot exp(-\frac{1}{2}((Ax + b - y)^T A^{-T} A^T \Sigma_y^{-1} A A^{-1} (Ax + b - y) + (x - \mu_x)^T \Sigma_x^{-1} (x - \mu_x)) \tag{342}$$

$$= \int_{-\infty}^{\infty} C \cdot exp(-\frac{1}{2}((x - A^{-1}(y - b))^T A^T \Sigma_y^{-1} A (x - A^{-1}(y - b)) + (x - \mu_x)^T \Sigma_x^{-1} (x - \mu_x)) \tag{343}$$

$$= \int_{-\infty}^{\infty} \mathcal{N}(x; A^{-1}(y - b), A^{-T} \Sigma_y A^{-1}) \mathcal{N}(x; \mu_x, \Sigma_x) dx \tag{344}$$

$$= \mathcal{N}(A^{-1} y; \mu_x + A^{-1} b, A^{-T} \Sigma_y A^{-1} + \Sigma_x) \tag{345}$$

$$= \mathcal{N}(y; A\mu_x + b, \Sigma_y + A^T \Sigma_x A) \tag{346}$$

We believe this can be applied to deal correctly with the relationship between dS and dS^O in Section 4.3. Currently, we worked with certain observations, while in reality observations contain some additional uncertainty. The equations in this section show that whenever we have a Gaussian random variable of which the mean is an affine transformation of a Gaussian, the uncertainty of this Gaussian can be moved to the covariance matrix of the original Gaussian.

C.4 Marginal Gaussian inference equations

In this section of the appendix, we will show equations that can be used to infer conditional distributions for guassians in the following setting:

$$Y = AZ + b \tag{347}$$

$$p_Z(z) = \mathcal{N}(z; X, \Sigma_z) \tag{348}$$

$$p_X(x) = \mathcal{N}(x; \mu_x, \Sigma_x) \tag{349}$$

$$p_Y(y|X = x) = \mathcal{N}(y; Ax + b, \Sigma_y) \tag{350}$$

We get from (Bishop, 2006, p. 93) that:

$$p_Y(y) = \mathcal{N}(y; A\mu_x, \Sigma_y + A\Sigma_x A^T) \tag{351}$$

$$p_X(x|Y = y) = \mathcal{N}\left(x; \Sigma_x^{\text{new}} \left(A^T \Sigma_y^{-1} y + \Sigma_x^{-1} \mu_x\right), \Sigma_x^{\text{new}}\right) \tag{352}$$

$$\Sigma_x^{\text{new}} = \left(\Sigma_x^{-1} + A\Sigma_y^{-1} A^T\right)^{-1} \tag{353}$$

These equations are the exact conditional distributions, therefore they can be utilized to calculate the minimum MSE estimators for μ_x and Σ_x. The method does not calculate Σ_z, this matrix is chosen or calculated separately.

C.5 Convergence and asymptotic unbiasedness of conditional Gaussian

In this section, we will look at the convergence properties of the method proposed in Section 5.6. We make use of matrix manipulations that are listed in (Petersen & Pedersen, 2012, the Matrix

Cookbook). We repeat the equations with an alternative notation:

$$Y = A\hat{Z} + b \tag{354}$$
$$p_Z(z) = \mathcal{N}(z; \hat{X}, \hat{\Sigma}_z) \tag{355}$$
$$p_W(x) = \mathcal{N}(x; \hat{\mu}_x, \hat{\Sigma}_x) \tag{356}$$
$$p_Y(y|X = x) = \mathcal{N}(y; Ax + b, A\hat{\Sigma}_z A^T) \tag{357}$$

For the error of the estimate $\hat{\mu}_x$ of μ_x we now get the following equations:

$$E[\mu_x - \hat{\mu}_x] = \epsilon_\mu \tag{358}$$
$$\|E[\mu_x - \hat{\mu}_x]\| = \|\epsilon_\mu\| \tag{359}$$

We now get that:

$$p_Y(y) = \mathcal{N}(y; A\hat{\mu}_x + b, A(\hat{\Sigma}_z + \hat{\Sigma}_x)A^T) \tag{360}$$
$$p_X(x|\hat{Y} = y) = \mathcal{N}(x; \Sigma_x^{new}(A^T(A\hat{\Sigma}_z A^T)^{-1}(y - b) + \hat{\Sigma}_x^{-1}\hat{\mu}_x), \hat{\Sigma}_x^{new}) \tag{361}$$
$$\hat{\mu}_x^{new} = \Sigma_x^{new}(A^T(A\hat{\Sigma}_z A^T)^{-1}(y - b) + \hat{\Sigma}_x^{-1}\hat{\mu}_x) \tag{362}$$
$$\hat{\Sigma}_x^{new} = (\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)^{-1} \tag{363}$$
$$E[\hat{\mu}_x^{new} - \mu_x] = (\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)^{-1}(A^T(A\hat{\Sigma}_z A^T)^{-1}A\mu_x + \hat{\Sigma}_x^{-1}(\mu_x + \epsilon_\mu)) - \mu_x \tag{364}$$
$$= (\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)^{-1}\hat{\Sigma}_x^{-1}\epsilon_\mu \tag{365}$$
$$\mu_x = (\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)^{-1}(\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)\mu_x \tag{366}$$
$$\|E[\hat{\mu}_x^{new} - \mu_x]\| = \|(\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)^{-1}\hat{\Sigma}_x^{-1}\epsilon_\mu\| \tag{367}$$
$$\leq \|(\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)^{-1}\hat{\Sigma}_x^{-1}\|\ \|\epsilon_\mu\| \tag{368}$$
$$\leq \ \|\epsilon_\mu\| \tag{369}$$

To go from Equation (368) to (369) we use the following:

$$\|(C + B)^{-1}C\| = \|(C + B)^{-1}(C^{-1})^{-1}\| \tag{370}$$
$$= \|(C^{-1}(C + B))^{-1}\| \tag{371}$$
$$= \|(I + C^{-1}B)^{-1}\| \tag{372}$$

We call $\lambda(K)$ the set of eigenvalues of the matrix K, and indicate a specific eigenvalue by $\lambda_i \in lambda(K)$. We note that in our case matrices C and B represent:

$$C = \hat{\Sigma}_x^{-1} \tag{373}$$
$$B = A^T(A\hat{\Sigma}_z A^T)^{-1}A \tag{374}$$

Both matrices are symmetric and positive semi-definite. Therefore we have that $\lambda(C) > 0$ and $\lambda(B) > 0$, i.e. all the eigenvalues of C and B are positive. We use that $\lambda_i \in \lambda(C) \implies \frac{1}{\lambda_i} \in \lambda(C^{-1})$ from (Petersen & Pedersen, 2012, eq. 287), therefore $\lambda(C^{-1}) > 0$. We now use $\lambda(FD) = \lambda(DF)$ and $\lambda_i \in \lambda(D) \implies (1 + \lambda_i) \in \lambda(I + D)$ from (Petersen & Pedersen, 2012, eq. 280 & 285):

$$\lambda_{\min}(C^{-1}B) = \lambda(C^{-1}B^{\frac{1}{2}}B^{\frac{1}{2}}) \tag{375}$$
$$= \lambda(B^{\frac{1}{2}}C^{-1}B^{\frac{1}{2}}) \tag{376}$$
$$\min \lambda(C^{-1}B) = \operatorname*{argmin}_\lambda \frac{(B^{\frac{1}{2}}C^{-1}B^{\frac{1}{2}}x, x)}{(x, x)} \tag{377}$$
$$= \operatorname*{argmin}_\lambda \frac{(C^{-1}B^{\frac{1}{2}}x, B^{\frac{1}{2}}x)}{(B^{\frac{1}{2}}x, B^{\frac{1}{2}}x)} \frac{(B^{\frac{1}{2}}x, B^{\frac{1}{2}}x)}{(x, x)} \tag{378}$$
$$\geq \lambda_{\min}(C^{-1})\lambda_{\min}(B) \geq 0 \tag{379}$$
$$\lambda_{\min}(I + C^{-1}B) \geq 1 \tag{380}$$
$$0 \leq \lambda((I + C^{-1}B)^{-1}) \leq 1 \tag{381}$$
$$\|(I + C^{-1}B)^{-1}\| \leq 1 \tag{382}$$

Finally, we use that $B^{\frac{1}{2}}$ exists, as every PSD matrix can be decomposed as $B = VDV^T$, where V is an orthonormal matrix that contains all the eigenvectors and D is a diagonal matrix that contains the corresponding eigenvalues. We can see that $B^{\frac{1}{2}} = VD^{\frac{1}{2}}V^T$, where $D^{\frac{1}{2}}$ is a diagonal matrix with the square root of all corresponding eigenvalues. We conclude that:

$$\|(\hat{\Sigma}_x^{-1} + A^T(A\hat{\Sigma}_z A^T)^{-1}A)^{-1}\hat{\Sigma}_x^{-1}\epsilon_\mu\| \leq \|\epsilon_\mu\| \tag{383}$$

This means that the bias in our estimate decreases after every iteration, therefore our estimator is asymptotically unbiased.

C.6 Bayesian Inference of a Multivariate Normal Distribution

In this section, we will introduce the general Bayesian inference method for multivariate Gaussian Distributions. We will apply the problem to our specific context in Appendix C.7. Firstly we start out with the prior probability distribution over the player qualities Q and the player inconsistency matrix Σ_q:

$$p_Q\left(q^O|\mu_q = M, \Sigma_q = S\right) = \mathcal{N}\left(q^O; M, S\right) \tag{384}$$
$$p_{\mu_q}\left(M|\Sigma_q = S\right) = \mathcal{N}\left(M; \mu_0, \beta_0 S\right) \tag{385}$$
$$p_{\Sigma_q}\left(S\right) = \mathcal{W}\left(S; a_0, B_0\right) \tag{386}$$
$$E\left[\Sigma_q\right] = a_0 B_0^{-1} \tag{387}$$
$$a_0 \in \mathbb{R}, \beta_0 \in \mathbb{R}, \mu_0 \in \mathbb{R}^{N_{PQ}}, B_0 \in \mathbb{R}^{N_{PQ}\times N_{PQ}} \tag{388}$$

Here $\mathcal{W}$ denoted the Weibull distribution. The idea is that we have N observations of all the player qualities, $q_i^O \sim Q$, and we use all these to get the following posterior distributions:

$$p_{\mu_q}(M|\Sigma_q = S, Q = q^O) = \mathcal{N}(M; \mu_N, \beta_N \Sigma_q) \tag{389}$$
$$p_{\Sigma_q}(S; Q = q^O) = \mathcal{W}(S; a_N, B_N) \tag{390}$$

Here we have that:

$$\mu_N = \frac{\beta_0\mu_0 + N\bar{q}}{\beta_N} \tag{391}$$
$$\beta_N = \beta_0 + N \tag{392}$$
$$a = a_0 + \frac{N}{2} \tag{393}$$
$$B_N = B_0 + \frac{N}{2}\left[\bar{\Sigma} + \frac{\beta_0}{\beta_N}(\bar{q} - \mu_0)(\bar{q} - \mu_0)^T\right] \tag{394}$$
$$\bar{q} = \frac{1}{N}\sum_{n=1}^{N} q_n^O \tag{395}$$
$$\bar{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(q_n^O - \bar{q})(q_n^O - \bar{q})^T \tag{396}$$

C.7 Application of Bayesian Inference with unknown partial observations

Throughout the project, we have tried to apply Bayesian Inference as a methodology for player rating parameter estimates. In this section, we will outline the progress we made with applying the method to our specific problem. The approach we take is very similar to the approach to (Glickman, 1993), and we use the general method of Bayesian Inference for Gaussians was taken from (Penny, 2014). We have rewritten this general method in Section C.6 according to the notation we have used throughout this thesis.

Before we can apply the general method to our problem, we identified crucial differences that require modifications in the algorithm. Firstly; the general Bayesian Inference for Gaussians algorithm assumes complete observations of the vectors that need to be estimated, while we only observe a subset of players during a match. We need to modify the algorithm so that it can deal with "importance weighted" observations. We will do this by changing a, β and N to diagonal matrices, containing the appropriate inputs, specific for each player-quality.

Secondly, we must apply inference incrementally. This means that we do batch processing, rather

than processing the complete dataset at once. This is not difficult, during each batch we define its outcome as the posterior, which will be considered the prior for the next batch.
Lastly; we currently do not have any observations of player performance, only observations of outcome implied strength difference between teams. An additional difficulty is that this strength difference contains some uncertainty. We solve this by estimating player performance from the implied strength difference.
As discussed earlier, we will replace a, β and N, by matrices we will call A, B and D. We see the following equations:

$$B_N^t = diag(\beta_1^t, \beta_2^t, ..., \beta_P^t) \tag{397}$$

$$= \begin{bmatrix} \beta_1 & & & \\ & \beta_2 & & \\ & & ... & \\ & & & \beta_{N_{PQ}} \end{bmatrix} \tag{398}$$

$$A_N^t = diag(a_1^t, a_2^t, ..., a_{N_{PQ}}^t) \tag{399}$$

$$D = diag(\delta^+ \mathbb{1}_P) \tag{400}$$

$$(\delta^+)_{ij} = |(\delta)_{ij}| \tag{401}$$

D is defined as the total amount of observations, but in our case we observe only a subset of players, not all with same importance. We will count a fully played match as one observation. An observation is irrelevant of the sign of the coefficient, therefore we take the column-sums of δ^+. which equals to δ but we take the absolute value of all elements.
To be able to apply the Bayesian methodology, we must have observations of player performances. Unfortunately, we only have outcomes that we can translate to an observed strength difference. To get a useful player quality observation we can apply the same procedure as in 5.6. Whereas the method initially was used to infer information about the player quality mean, we will now simply infer the player performance during a single match. We use $\Sigma_{dS|q^O}$, the covariance matrix of strength difference if the strength difference would be known. As q^O has a direct mapping to match outcome, we can use Equation (144) to determine $\Sigma_{dS|q^O}$.
We get the following prior equations:

$$P(q^O|\mu_q, \Sigma_q) = \mathcal{N}(q^O; \mu_q, \Sigma_q) \tag{402}$$

$$P(dS|\mu_q, \Sigma_q, q^O) = \mathcal{N}(dS; \delta q^O; \Sigma_{dS|q^O}) \tag{403}$$

Which leads to the following posterior probability distributions, using (Bishop, 2006):

$$P(dS|\mu_q, \Sigma_q) = \mathcal{N}(dS; \delta\mu_q, \Sigma_{dS|q^O} + \delta\Sigma_q\delta^T) \tag{404}$$

$$P(q^O|\mu_q, \Sigma_q, dS) = \mathcal{N}(q^O; J(\delta^T(\Sigma_{dS|q^O})^{-1}dS + \Sigma_q^{-1}\mu_q), J) \tag{405}$$

$$J = (\Sigma_q + \delta^T\Sigma_{dS|q^O}\delta)^{-1} \tag{406}$$

Now we can put the notation in block-matrix form. For readability, we assume that, index-wise, all active players are first, and inactive player are after that. This can be achieved by specific columns (and the same row) permutations. We get the following equations:

$$\begin{bmatrix} \mu_t^{(1)} \\ \mu_t^{(2)} \end{bmatrix} = \gamma_t^{-1} \left(\gamma_{t-1} \begin{bmatrix} \mu_{t-1}^{(1)} \\ \mu_{t-1}^{(2)} \end{bmatrix} + \begin{bmatrix} D_t & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mu_{q^t}^{(1)} \\ \mu_{q^t}^{(2)} \end{bmatrix} \right) \tag{407}$$

$$\gamma_t = \gamma_{t-1} + \begin{bmatrix} D_t & 0 \\ 0 & 0 \end{bmatrix} \tag{408}$$

$$A_t = A_{t-1} + \frac{1}{2} \begin{bmatrix} D_t & 0 \\ 0 & 0 \end{bmatrix} \tag{409}$$

$$B_t = B_{t-1} + K = B_{t-1} + \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \tag{410}$$

$$K_{11} = \frac{1}{2} D_t \left[\bar{\Sigma}^t + \gamma_{t-1}(\gamma_t)^{-1}(\bar{q} - \mu^{t-1})(\bar{q} - \mu^{t-1})^T \right] \tag{411}$$

$$K_{12} = K_{21}^T = 0 \tag{412}$$

$$K_{22} = 0 \tag{413}$$

$$\bar{q}^t = \mu_{q^O} = J(\delta^T(\Sigma_{dS|q^O})^{-1}dS + \Sigma_q^{-1}\mu_q) \text{ from Equation (405)} \tag{414}$$

$$\bar{\Sigma}^t = \Sigma_{q^O} = J = (\Sigma_q + \delta^T\Sigma_{dS|q^O}\delta)^{-1} \text{ from Equation (406)} \tag{415}$$

For the players where we have an observation, we know how to update B_t. Unfortunately, we have not found how to do this exactly for the other parts of K, we have chosen the $K_{12} = K_{21}^T$ to be empty. With the current approach, we are certain B_t remains PSD symmetric matrix, because we only add K which is a PSD symmetric matrix.

D Heteroskedastic player inconsistency

In all the models discussed in Section 5 we assumed that player performance variance was equal for all players; we assumed homoskedasticity. In this section of the appendix, we will look at methods that can estimate the player variances under the assumption that these are uncorrelated and player specific, thus different for each player; we call this heteroskedastic. We did not implement any methods discussed in this section due to computational difficulty and estimator uncertainty. We believe more data points per player are needed in order to be able to estimate the specific inconsistency parameter for each player-quality.
We will assume in this section that we have an estimator for μ_{M_q}, which we use to calculate the residuals ϵ. If we apply the OLS-estimation procedure we get:

$$\Sigma_{dS} = (\delta \Sigma_q \delta^T \circ I) \tag{416}$$
$$\hat{\Sigma}_{M_q} = (\delta^T\delta)^{-1}\delta^T(\hat{\Sigma}_{dS})\delta(\delta^T\delta)^{-1} \tag{417}$$
$$\epsilon = dS - \hat{dS} \tag{418}$$
$$= dS - \delta\hat{\mu}_{M_q} \tag{419}$$
$$= (I - \delta(\delta^T\delta)^{-1}\delta^T)dS \tag{420}$$
$$\epsilon_k \sim \mathcal{N}(0, \sum_{i=1}^{N_{PQ}} \delta_{k,i}^2(\Sigma_q)_{i,i}) = \mathcal{N}(0, \sum_{i=1}^{N_{PQ}} \delta_{k,i}^2\sigma_{q_i}^2) \tag{421}$$

So we have an expression for the estimated error ϵ in equation (420) and we know its distribution to be as in equation (421). We see that we can write these equations for all matches as:

$$\begin{aligned}
\text{Var}(\epsilon_1) &= \delta_{1,1}^2\sigma_{q_1}^2 + \delta_{1,2}^2\sigma_{q_2}^2 + \ldots + \delta_{1,N_{PQ}}^2\sigma_{q_{N_{PQ}}}^2 \\
\text{Var}(\epsilon_2) &= \delta_{2,1}^2\sigma_{q_1}^2 + \delta_{2,2}^2\sigma_{q_2}^2 + \ldots + \delta_{2,N_{PQ}}^2\sigma_{q_{N_{PQ}}}^2 \\
&\ldots \\
\text{Var}(\epsilon_{N_{KM}}) &= \delta_{N_{KM},1}^2\sigma_{q_1}^2 + \delta_{N_{KM},2}^2\sigma_{q_2}^2 + \ldots + \delta_{N_{KM},N_{PQ}}^2\sigma_{q_{N_{PQ}}}^2
\end{aligned}$$

These equations can be represented conveniently in matrix form, yielding the relationship between observed errors and our estimator for the player inconsistency of player qualities:

$$E\begin{bmatrix}\epsilon_1^2 \\ \epsilon_2^2 \\ \ldots \\ \epsilon_{N_{KM}}^2\end{bmatrix} = \begin{bmatrix}\delta_{1,1}^2 & \delta_{1,2}^2 & \ldots & \delta_{1,N_{PQ}}^2 \\ \delta_{2,1}^2 & \delta_{2,2}^2 & \ldots & \delta_{2,N_{PQ}}^2 \\ \ldots & \ldots & \ldots & \ldots \\ \delta_{N_{KM},1}^2 & \delta_{N_{KM},2}^2 & \ldots & \delta_{N_{KM},N_{PQ}}^2\end{bmatrix}\begin{bmatrix}\sigma_{q_1}^2 \\ \sigma_{q_2}^2 \\ \ldots \\ \sigma_{q_{N_{PQ}}}^2\end{bmatrix} \tag{422}$$
$$E[\epsilon^{\circ 2}] = \delta^{\circ 2}\vec{\sigma}^{\circ 2} \tag{423}$$
$$= \begin{bmatrix}\delta_{1,1}^2\sigma_{q_1} & \delta_{1,2}^2\sigma_{q_2} & \ldots & \delta_{1,N_{PQ}}^2\sigma_{q_{N_{PQ}}} \\ \delta_{2,1}^2\sigma_{q_1} & \delta_{2,2}^2\sigma_{q_2} & \ldots & \delta_{2,N_{PQ}}^2\sigma_{q_{N_{PQ}}} \\ \ldots & \ldots & \ldots & \ldots \\ \delta_{N_{KM},1}^2\sigma_{q_1} & \delta_{N_{KM},2}^2\sigma_{q_2} & \ldots & \delta_{N_{KM},N_{PQ}}^2\sigma_{q_{N_{PQ}}}\end{bmatrix}\begin{bmatrix}\sigma_{q_1} \\ \sigma_{q_2} \\ \ldots \\ \sigma_{q_{N_{PQ}}}\end{bmatrix} \tag{424}$$
$$= \left(\delta^{\circ 2} \circ (\vec{\sigma}^T \otimes \vec{\mathbb{1}}_{N_{KM}})\right)\vec{\sigma} \tag{425}$$

The two formulations can be used to find appropriate estimators $\hat{\sigma}_{q_k}$ by minimizing the difference between the left and righthand side of equation (422) or (424). This difference can be defined in different ways, we will explore squared, absolute difference and relative difference in sections D.2, D.3 and D.4 respectively.
Another applicable method we found in literature is called variance component analysis. The idea of this method is to break up the total variance over the components. In our case, components are individual player qualities, and we write Σ_q as a sum of matrices multiplied by the variances of

individual players:

$$\Sigma_{dS} = \sum_{i=1}^{N_{PQ}} \sigma_{q_i}^2 R_{q_i} \tag{426}$$

$$R_{q_i} = \begin{bmatrix} \delta_{1,i}^2 & 0 & \dots & 0 \\ 0 & \delta_{2,i}^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \delta_{N_{KM},i}^2 \end{bmatrix} \tag{427}$$

D.1 Heteroskedasticity - maximum likelihood estimation

A method that is used often to estimate parameters is Maximum Likelihood Estimation. The idea is that we find estimators that maximize the likelihood of our observations. We have attempted to do this for all parameters using the observations in 5.1. In this subsection, we will start with the MLE equations and derive an expression for the maximum likelihood estimators:

$$\mathcal{L}(\hat{\Sigma}_q, \hat{\mu}_{M_q}; \epsilon = \hat{\epsilon}) = P_{\hat{\Sigma}_q, \hat{\mu}_{M_q}}(\epsilon = \hat{\epsilon}) \tag{428}$$

$$= \left(\frac{1}{2\pi}\right)^{\frac{1}{2}N_{KM}} \frac{1}{\sqrt{|\delta\hat{\Sigma}_q\delta^T \circ I|}} \exp\left(-\frac{1}{2}\hat{\epsilon}\left(\delta\hat{\Sigma}_q\delta^T \circ I\right)^{-1}\hat{\epsilon}^T\right) \tag{429}$$

$$= \left(\frac{1}{2\pi}\right)^{\frac{1}{2}N_{KM}} \left(\prod_{j=1}^{M}\left(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2\right)^{-\frac{1}{2}}\right) \exp\left(-\frac{1}{2}\sum_{j=1}^{M}\hat{\epsilon}_j^2(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2)^{-1}\right) \tag{430}$$

$$\log P_{\hat{\Sigma}_q, \hat{\mu}_{M_q}}(\epsilon = \hat{\epsilon}) = -\frac{1}{2}N_{KM}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{M}\log\left(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2\right) - \frac{1}{2}\sum_{j=1}^{M}\hat{\epsilon}_j^2(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2)^{-1} \tag{431}$$

$$\frac{\partial \log P_{\hat{\Sigma}_q, \hat{\mu}_{M_q}}(\epsilon = \hat{\epsilon})}{\partial \sigma_{q_k}} = -\frac{1}{2}\sum_{j=1}^{M}\frac{2\delta_{j,k}^2\sigma_{q_k}}{\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2} + \frac{1}{2}\sum_{j=1}^{M}\hat{\epsilon}_j^2 2\delta_{j,k}^2\sigma_{q_k}\left(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2\right)^{-2} \tag{432}$$

$$= \sum_{j=1}^{M}\left(\left(\hat{\epsilon}_j^2\left(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2\right)^{-1} - 1\right)\delta_{j,k}^2\sigma_{q_k}\left(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2\right)^{-1}\right) \tag{433}$$

$$\frac{\partial \log P_{\hat{\Sigma}_q, \hat{\mu}_{M_q}}(\epsilon = \hat{\epsilon})}{\partial (\hat{\mu}_{M_q})_h} = \sum_{j=1}^{M}\frac{\partial \log P(\hat{\epsilon}|\hat{\Sigma}_q, \hat{\mu}_{M_q})}{\partial \hat{\epsilon}_j}\frac{\partial \hat{\epsilon}_j}{\partial (\hat{\mu}_{M_q})_h} \tag{434}$$

$$= -\sum_{j=1}^{M}\hat{\epsilon}_j(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2)^{-1}\cdot(-\delta_{j,h}) \tag{435}$$

$$= \sum_{j=1}^{M}\hat{\epsilon}_j\delta_{j,h}(\sum_{i=1}^{N}\delta_{j,i}^2\hat{\sigma}_{q_i}^2)^{-1} \tag{436}$$

We can now solve $\frac{\partial \log P_{\hat{\Sigma}_q, \hat{\mu}_{M_q}}(\epsilon=\hat{\epsilon})}{\partial (\hat{\mu}_{M_q})_h} = 0$ and $\frac{\partial \log P_{\hat{\Sigma}_q, \hat{\mu}_{M_q}}(\epsilon=\hat{\epsilon})}{\partial \hat{\sigma}_{q_k}} = 0$. These equations can only be solved numerically, and correspond to the equations in (Pelgrin, 2016).

D.2 Heteroskedasticity - least squares

We can use the Ordinary and Generalized Least Squares methods discussed in 5.2 and 5.3. We get from Equation (424):

$$\hat{\Sigma}_q^{H-OLS} = diag((\Delta^T\Delta)^{-1}\Delta^T\epsilon^2) \tag{437}$$

$$\hat{\Sigma}_q^{H-GLS} = diag((\Delta^T P\Delta)^{-1}\Delta^T P\epsilon^2) \tag{438}$$

$$\Delta = \delta^{\circ 2} \tag{439}$$

We can also use the second definition, here we use Equation (422) and we must iteratively solve the following equations until convergence:

$$\hat{\Sigma}_q^{H2-OLS} = diag((\Gamma^T\Gamma)^{-1}\Gamma^T\epsilon^2) \tag{440}$$

$$\hat{\Sigma}_q^{H2-GLS} = diag((\Gamma^T P\Gamma)^{-1}\Gamma^T P\epsilon^2) \tag{441}$$

$$\Gamma = \delta^{\circ 2} \circ (diag(\sqrt{\hat{\Sigma}_q}) \otimes \mathbb{1}_{N_{KM}}) \tag{442}$$

Even though the OLS and GLS estimators are unbiased, these estimates can produce undesirable results. A large drawback of this approach is that the estimator $\hat{\sigma}^2_{q_i}$ for a single player can be negative, while obviously $\sigma^2_{q_i}$ must be positive.
A possible solution to avoid negative (or very small) estimates of $\sigma^2_{q_i}$ we could apply the Non-Negative Least Squares (NNLS) method (Lawson & Hanson, 1995). The method has the same minimization criterion as OLS and GLS, but with an inequality constraint:

$$\hat{\Sigma}_q^{H-NNLS} = \underset{\sigma^{\circ 2}}{\operatorname{argmin}} \left(\epsilon - \Delta\vec{\sigma}^{\circ 2}\right)^2 \quad \text{subject to } \vec{\sigma}^{\circ 2} \geq C \tag{443}$$

To achieve a non-negative solution, we choose $C = 0$. An efficient algorithm to yield a solution of equation 443 is an an active set modification of the standard least squares model, and is described in (Lawson & Hanson, 1995, p. 161).

D.3 Heteroskedasticity - p-norm difference

Another way we can estimate heteroskedastic errors by observing that:

$$e_j \sim \mathcal{N}(0, \sum_{k=1}^{N_{PQ}} \delta^2_{j,k}\sigma^2_{q_k}) \tag{444}$$

$$E[e_j^2] = \sum_{k=1}^{N_{PQ}} \delta^2_{j,k}\sigma^2_{q_k} \tag{445}$$

$$\hat{\sigma}^{DEV_p} = \underset{\sigma}{\operatorname{argmin}} \sum_{j=1}^{N_{KM}} \left| e_j^2 - \sum_{k=1}^{N_{PQ}} \delta^2_{j,k}\sigma^2_{q_k} \right|^p \tag{446}$$

This estimator can be found for any $p > 0$ using a method called iteratively reweighted least squares (Burrus, 2012). This method uses the following equations:

$$\hat{\sigma}^{IRLS} = \underset{\sigma}{\operatorname{argmin}} \sum_{j=1}^{N_{KM}} w_j(\sigma) \left| e_j^2 - \sum_{k=1}^{N_{PQ}} \delta^2_{j,k}\sigma^2_{q_k} \right|^2 \tag{447}$$

$$w_j(\sigma) = \left| e_j^2 - \sum_{k=1}^{N_{PQ}} \delta^2_{j,k}\sigma^2_{q_k} \right|^{p-2} \tag{448}$$

After each iteration, the terms w_j are updated and the least squares problem in Equation (447) is solved again.

D.4 Heteroskedasticity - relative error

In the previous sections, we have estimated the player quality inconsistencies by looking at a norm of the differences between measured and expected error. Another possibility is to look at the relative error; for every j we have that:

$$E[e_j^2] = \sum_{i=1}^{N_{PQ}} \delta^2_{ji}\sigma^2_{q_i} \implies \frac{E[e_j^2]}{\sum_{i=1}^{N_{PQ}} \delta^2_{ji}\sigma^2_{q_i}} = 1 \tag{449}$$

We can take the natural logarithm, yielding:

$$\ln(E[e_j^2]) = \ln\left(\sum_{i=1}^{N_{PQ}} \delta^2_{ji}\sigma^2_{q_i}\right) \tag{450}$$

We would now seek the estimator for σ such that:

$$\hat{\sigma}^{MRE} = \underset{\sigma}{\operatorname{argmin}} \sum_{j=1}^{N_{KM}} \left| \ln(e_j^2) - \ln\left(\sum_{k=1}^{N_{PQ}} \delta_{j,k}^2 \sigma_{q_k}^2 \right) \right| \tag{451}$$

Unfortunately we cannot apply the same approach as in D.3, because our function of σ is non-linear. We can apply the Newton method for minimization (Murray, 2010). We define the function g as:

$$g(\sigma_q) = \sum_{j=1}^{N_{KM}} \left| \ln(e_j^2) - \ln\left(\sum_{i=1}^{N_{PQ}} \delta_{ji}^2 \sigma_{q_i}^2 \right) \right| \tag{452}$$

$$\frac{\partial g(\sigma_q)}{\partial \sigma_{q_k}} = 2 \sum_{j=1}^{N_{KM}} S_j(\sigma_q) \frac{\delta_{jk}^2 \sigma_{q_k}}{\sum_{i=1}^{N_{PQ}} \delta_{ji}^2 \sigma_{q_i}^2} \tag{453}$$

$$\frac{\partial^2 g(\sigma_q)}{\partial \sigma_{q_k}^2} = 2 \sum_{j=1}^{N_{KM}} S_j(\sigma_q) \left(\frac{\delta_{jk}^2}{\sum_{i=1}^{N_{PQ}} \delta_{ji}^2 \sigma_{q_i}^2} + \frac{\delta_{jk}^2 \sigma_{q_k}}{(\sum_{i=1}^{N_{PQ}} \delta_{ji}^2 \sigma_{q_i}^2)^2} \right) \qquad \text{where } S_j(\sigma) \neq 0 \ \ \forall_j \tag{454}$$

$$\frac{\partial^2 g(\sigma_q)}{\partial \sigma_{q_k} \partial \sigma_{q_m}} = 4 \sum_{j=1}^{N_{KM}} S_j(\sigma_q) \frac{\delta_{jk}^2 \sigma_{q_k} \delta_{jm}^2 \sigma_{q_m}}{(\sum_{i=1}^{N_{PQ}} \delta_{ji}^2 \sigma_{q_i}^2)^2} \qquad \text{where } S_j(\sigma) \neq 0 \ \ \forall_j \tag{455}$$

$$S_j(\sigma) = \operatorname{sign}\left(\ln(e_j^2) - \ln\left(\sum_{i=1}^{N_{PQ}} \delta_{ji}^2 \sigma_{q_i}^2 \right) \right) \tag{456}$$

The update equations become;

$$\hat{\sigma}^{new} = \hat{\sigma}^{old} - \left[\operatorname{Hess} g(\hat{\sigma}^{old}) \right]^{-1} \nabla g(\hat{\sigma}^{old}) \tag{457}$$

$$(\nabla f)_i \equiv \frac{\partial f}{\partial x_i} \tag{458}$$

$$(\operatorname{Hess} f)_{ij} \equiv \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{459}$$

When applying this method, we must consider that the factors $S_j(\sigma)$ are non-continuous. We did not have enough time to fully investigate this behavior, but we believe this method can be applied to yield an estimator under the criterion of Equation (451).

D.5 Heteroskedasticity - almost unbiased estimator

Another way to estimate the heteroskedastic variances is to apply variance component estimation techniques. There are multiple methods described by (Teunissen & Amiri-Simkooei, 2008). Such methods require a representation like in equations (426) and (427). Most methods require numerically expensive computations. We chose to consider a promising method that is efficient, biased but asymptotically unbiased, also described by (Rao, 1970). The estimator we yield is referred to as

almost unbiased estimator.

$$E[\epsilon^T \Sigma_{dS}^{-1} \epsilon] = Tr(\Sigma_{dS}^{-1} \Sigma_\epsilon) + E[\epsilon]^T \Sigma_{dS}^{-1} E[\epsilon] \tag{460}$$

$$E[\epsilon^T \Sigma_{dS}^{-1} \Sigma_{dS} \Sigma_{dS}^{-1} \epsilon] = Tr\left(\Sigma_{dS}^{-1}(I - \delta(\delta^T\delta)^{-1}\delta^T)\Sigma_{dS}\right) \tag{461}$$

$$\Sigma_{dS} = \sum_{i=1}^{N_{PQ}} \sigma_{q_i}^2 R_{q_i} \tag{462}$$

$$\sum_{i=1}^{N_{PQ}} E[\epsilon^T \Sigma_{dS}^{-1}(\sigma_{q_i}^2 R_{q_i})\Sigma_{dS}^{-1}\epsilon] = \sum_{i=1}^{N_{PQ}} Tr\left(\Sigma_{dS}^{-1}(I - \delta(\delta^T\delta)^{-1}\delta^T)\sigma_{q_i}^2 R_{q_i}\right) \tag{463}$$

$$\sigma_{q_k}^2 = \frac{E[\epsilon^T \Sigma_{dS}^{-1}(\sigma_{q_k}^2 R_{q_k})\Sigma_{dS}^{-1}\epsilon]}{Tr(\Sigma_{dS}^{-1}(I - \delta(\delta^T\delta)^{-1}\delta^T)R_{q_k})} \tag{464}$$

$$= E[\epsilon^T \Sigma_{dS}^{-1} R_{q_k} \Sigma_{dS}^{-1}\epsilon] \frac{\sigma_{q_k}^2}{Tr(R_{q_k}\Sigma_{dS}^{-1}(I - \delta(\delta^T\delta)^{-1}\delta^T))} \tag{465}$$

$$\Sigma_{dS}^{-1} R_{q_k} \Sigma_{dS}^{-1} = \begin{bmatrix} \frac{\delta_{k,1}^2}{(\sum_i \delta_{i,1}^2 \sigma_{i,N_1}^2)^2} & 0 & ... & 0 \\ 0 & \frac{\delta_{k,2}^2}{(\sum_i \delta_{i,2}^2 \sigma_{i,N_2}^2)^2} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \frac{\delta_{k,N_{PQ}}^2}{(\sum_i \delta_{i,N_{PQ}}^2 \sigma_{i,N_{PQ}}^2)^2} \end{bmatrix} \tag{466}$$

$$(\hat{\sigma}_{q_k}^{AUE})^2 = \left(\sum_{j=1}^{N_{PQ}} \frac{\delta_{k,j} e_j^2}{(\sum_i \delta_{i,j}\sigma_{i,j}^2)^2}\right) \frac{\sigma_{q_k}^2}{Tr(R_{q_k}\Sigma_{dS}^{-1}(I - \delta(\delta^T\delta)^{-1}\delta^T))} \tag{467}$$

To go from (464) to (465) we use that $Tr(ABC) = Tr(BCA) = Tr(CAB)$, from (Petersen & Pedersen, 2012). The trace in the denominator of expression (467) cannot be simplified further because $\delta(\delta^T\delta)^{-1}\delta^T$ is not a diagonal matrix. The idea is that if we apply this method iteratively, our estimation converges to an unbiased estimate.

References

Albert, James H., & Chib, Siddhartha. 1993. Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American Statistical Association*, **88**(422), 669–679.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Bradley, R., & Terry, M. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, **39**(3/4), 324–345.

Bromley, P.A. 2014. *Products and Convolutions of Gaussian Probability Density Functions*. `http://www.tina-vision.net/docs/memos/2003-003.pdf`. Internal Report.

Brooks, Joel, Kerr, Matthew, & Guttag, John. 2016. Developing a Data-Driven Player Ranking in Soccer Using Predictive Model Weights. *Pages 49–55 of: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: ACM.

Burkardt, John. 2004. *The Truncated Normal Distribution*. `https://people.sc.fsu.edu/~jburkardt/presentations/truncated_normal.pdf`. Online PDF Article.

Burrus, C. Sidney. 2012. *Iterative Reweighted Least Squares*. `https://cnx.org/exports/92b90377-2b34-49e4-b26f-7fe572db78a1@12.pdf/iterative-reweighted-least-squares-12.pdf`. Online PDF article.

Casalicchio, Giuseppe. 2013. *Modelling Comparison Data with Ordinal Response*. Ph.D. thesis, Ludwig Maximilians Universität München.

Crooks, Gavin E. 2013. *Logistic approximation to the logistic-normal integral, http://threeplusone.com/logistic-normal*.

Davidson, Roger R. 1970. On Extending the Bradley-Terry Model to Accommodate Ties in Paired Comparison Experiments. *Journal of the American Statistical Association*, **65**(329), 317–328.

Dendir, Seife. 2016. When do soccer players peak? A note. *Journal of Sports Analytics*, **2**(2), 89–105.

Elo, A.E. 1978. *The rating of chessplayers, past and present*. Arco Pub.

Fahrmeir, Ludwig, & Tutz, Gerhard. 1994. Dynamic stochastic models for time-dependent ordered paired comparison systems. *Journal of the American Statistical Association*, **89**(428), 1438–1449.

Falmagne, J.C. 1971. The generalized Fechner problem and discrimination. *Journal of Mathematical Psychology*, **8**(1), 22 – 43.

Fitzpatrick, Jamie. 2017. *What Is the Plus/Minus Statistic in Hockey and How Is It Calculated?, https://www.thoughtco.com/what-is-the-plus-minus-statistic-2779372*.

Gillies, Donald B. 1959. Solutions to general non-zero-sum games. *Contributions to the Theory of Games*, **4**(40), 47–85.

Gilovich, Thomas, Vallone, Robert, & Tversky, Amos. 1985. The hot hand in basketball: On the misperception of random sequences. *Cognitive Psychology*, **17**(3), 295 – 314.

Glickman, Mark E. 1993. *Paired Comparison Models with Time Varying Parameters*. Doctoral thesis, Harvard University Dept of Statistics.

Glickman, Mark E. 1999. Parameter estimation in large dynamic paired comparison experiments. *Applied Statistics*, **48**(3), 377–394.

Grinstead, C.M., & Snell, J.L. 2009. *Grinstead and Snell's Introduction to Probability, Chapter 7*. University Press of Florida.

Herbrich, Ralf, Minka, Tom, & Graepel, Thore. 2007 (January). TrueSkill(TM): A Bayesian Skill Rating System. *In: Microsoft Research*.

Hoerl, Arthur E., & Kennard, Robert W. 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, **12**(1), 55–67.

Hunter, David R. 2004. MM algorithms for generalized Bradley-Terry models. *Ann. Statist.*, **32**(1), 384–406.

Indatabet. 2017. *Football 2-in-1 ML-TG Dataset.* `https://www.indatabet.com/ft-2in1.html`. Online Data Aggregation Service.

King, Daniel. 1997. *Kasparov VS. Deeper Blue: The Ultimate Man VS. Machine Challenge.* Trafalgar Square.

Kitano, Hiroaki, Asada, Minoru, Kuniyoshi, Yasuo, Noda, Itsuki, & Osawa, Eiichi. 1997. *RoboCup: The Robot World Cup Initiative.*

KNTB. 2017. *"Dynamic Playing Strength" system (DSS) used by the KNTB (Dutch Tennis Federation) to rank all their players according to their playing level.* `http://www.knltb.nl/tennissers/speelsterkte/`.

Kullback, S., & Leibler, R. A. 1951. On Information and Sufficiency. *Ann. Math. Statist.*, **22**(1), 79–86.

Lawson, C.L., & Hanson, R.J. 1995. *Solving Least Squares Problems.* Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.

Likert, R. 1932. A technique for the measurement of attitudes. *Archives of Psychology*, **22**(140), 1–55.

MacKay, David J. C. 2002. *Information Theory, Inference & Learning Algorithms.* New York, NY, USA: Cambridge University Press.

McCullagh, Peter. 1980. Regression models for ordinal data. *Journal of the royal statistical society. Series B (Methodological)*, **42**(2), 109–142.

Mosteller, Frederick. 1951. Remarks on the method of paired comparisons: I. The least squares solution assuming equal standard deviations and equal correlations. *Psychometrika*, **16**(1), 3–9.

Murray, W. 2010. *Newton-type Methods.* `https://web.stanford.edu/class/cme334/docs/newton-type-methods.pdf`. Online PDF Article.

Nikolenko, Sergey, & Sirotkin, Alexander. 2011. A New Bayesian Rating System for Team Competitions. *Pages 601–608 of:* Getoor, Lise, & Scheffer, Tobias (eds), *Proceedings of the 28th International Conference on Machine Learning (ICML-11).* New York, NY, USA: ACM.

OpenAI. 2017. *OpenAI is a company that created the first bot that beat top human players in 1vs1 matches of the game Dota 2.* `https://blog.openai.com/dota-2/`.

Pelgrin, Florian. 2016 (9). *Heteroscedasticity.* `https://hec.unil.ch/docs/files/46/263/slides_chapter_5_part_ii.pdf`. Presentation.

Penny, Will. 2014. *Bayesian Inference for the Multivariate Normal.* `http://www.fil.ion.ucl.ac.uk/~wpenny/publications/bmn.pdf`. Online PDF Article.

Peters, Jan, Kober, Jens, Mülling, Katharina, Krämer, Oliver, & Neumann, Gerhard. 2013. *Towards Robot Skill Learning: From Simple Skills to Table Tennis.* Berlin, Heidelberg: Springer Berlin Heidelberg. Pages 627–631.

Petersen, K. B., & Pedersen, M. S. 2012. *The Matrix Cookbook.* `https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf`. Technical University of Denmark, Version 2012-11-15.

Press, William H., Teukolsky, Saul A., Vetterling, William T., & Flannery, Brian P. 1992. *Numerical Recipes: The Art of Scientific Computing (3rd ed.).* New York, NY, USA: Cambridge University Press.

Ranganathan, Ananth. 2004. *Assumed Density Filtering.* `https://pdfs.semanticscholar.org/3c35/80010130cc0ba35ce177221953168b30f5fa.pdf`. Online PDF Article.

Rao, C. Radhakrishna. 1970. Estimation of Heteroscedastic Variances in Linear Models. *Journal of the American Statistical Association*, **65**(329), 161–172.

Rao, C.R. 1973. *Linear statistical inference and its applications*. Vol. 2. Wiley New York.

Robocup. 2017. *RoboCup is an international scientific initiative with the goal to advance the state of the art of intelligent robots. When established in 1997, the original mission was to field a team of robots capable of winning against the human soccer World Cup champions by 2050.* `www.robocup.org`.

Rosenbaum, Dan T. 2004. *Measuring How NBA Players Help Their Teams Win,* `http://www.82games.com/comm30.htm`.

Shapley, Lloyd S. 1953. A value for n-person games. *Contributions to the Theory of Games*, **2**(28), 307–317.

Statista. 2012. *Statista Report, Total global online gambling gross win by category in 2012 (in billion euros).* `https://www.statista.com/statistics/253396/interactive-gambling-gross-win-worldwide-by-game-type/`.

Teunissen, P. J. G., & Amiri-Simkooei, A. R. 2008. Least-squares variance component estimation. *Journal of Geodesy*, **82**(2), 65–82.

Thurstone, L. L. 1927. A Law of Comparative Judgment. *Psychological Review*, **34**(4), 273–286.

Tiedemann, T, Francksen, T, & Latacz-Lohmann, Uwe. 2011. Assessing the performance of German Bundesliga football players: a non-parametric metafrontier approach. *Central European Journal of Operations Research*, **19**(4), 571–587.

Timmer, Judith, Boucherie, Richard J., Lammers, Esmé, Baër, Niek, Bos, Maarten, & Feenstra, Arjan. 2017. Estimating the potential of collaborating professionals, with an application to the Dutch film industry. *OR Spectrum*, Oct.