Modelling And Visualization Of A Bridge Player's Performance

Maciej Krajowski-Kukiel

Kongens Lyngby 2011
IMM-MSC-2011-78

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Summary

The thesis consists of two main parts. The first one is building a model of a bridge player's performance. The solution used is based on a popular rating system known as Elo. The idea is to express a skill level as a random variable following a Logistic Distribution. Since the rules and scoring in bridge are considerably complicated, an extensive amount of work has been put into improving the basic formulas and to reflect some real life dependencies in the model. The second part is a visualization of how the model works by applying it to real bridge players' statistics. In the final process $22,000,000$ games were considered for more than $200,000$ players. Two metrics have been used to verify the model's correctness and usefulness - Root Mean Square Error and Binomial Deviance. The latter one can be directly transformed into accuracy. The final rate of correct prediction is 50,0158%, which is a little better than the null model, with an accuracy of 50% (a model of 50/50 random outcome). Even though it is not an impressive score, it is considered a success. Comparing to chess, which is a two-player game and includes only one additional parameter - the person who started the game is more likely to win - it is much harder to predict the winner of a bridge game. The main reasons of this are that bridge is a game of chance, in one game many independent partnerships are involved, which results are compared to each other.
Several visualization techniques have been used to investigate what features of the obtained model behave correctly and which appear erroneous. It also helped in defining the problems, which are most likely to result in low accuracy. The analysis is summarized by creating a list of future tasks.
The obtained system in its current form is not optimal, however it gives some reasonable results and a proper base that can be extended in the future.

Resumé

Afhandlingen består af to overordnede dele. Den første er konstruktionen af en model af en bridgespillers præstation. Den valgte løsning er baseret på det populære rating-system kendt under betegnelsen Elo, som blev opfundet til skak i 1978. Den grundlæggende idé er at udtrykke graden af en spillers dygtighed som en tilfældig variabel ifølge Logistic Distribution. Da reglerne og pointsystemet i bridge er forholdsvis komplicerede, er der blevet brugt store ressourcer på at forbedre de grundlæggende formler og på at udtrykke nogle virkelige udfald af modellen. Afhandlingens anden del er en visualisering af hvordan modellen virker når den overføres til bridgespilleres data/statistik. I alt blev $22,000,000$ spil og $200,000$ spillere taget i betragtning. To måleudtryk er blevet brugt til at verificere modellens korrekthed og brugbarhed - Root Mean Square afvigelse(RMS error) og binomial afvigelse (binomial deviance). Sidstnævnte kan ses som et direkte udtryk for præcisionen af modellen. Den opnåede rate for korrekt forudsigelse udgør 50,0158%, som er lidt bedre end null-systemet, med en præcision på 50%(en model med 50/50 chance - tilfældigt udfald). Selvom det ikke er et imponerende resultat, kan det betragtes som en succes. Sammenlignet med skak, som er et to-personers spil og kun indeholder et enkelt yderligere parameter - at personen som har startet spillet, har en øget sandsynlighed for at vinde - er det væsentligt sværere at forudsige vinderen af et bridge-spil. Hovedsageligt er forskellen, at bridge er baseret delvis på tilfældighed, samt at i et enkelt spil er flere uafhængige partnerskaber involveret, hvis resultater sammenlignes.
Flere visualiseringsteknikker er blevet taget i brug for at undersøge hvilke egenskaber ved den eksisterende model som fungerer korrekt og hvilke som er fejlagtige. Dettte har også hjulpet med at definere de problemer, der har den største sandsynlighed for at resultere i en lav præcision. Analysen opsummeres

med en liste over fremtidige opgaver.
Det udviklede system er ikke optimalt i dets nuværende form, men det giver et sæt fornuftige resultater og en solid basis som kan udvides i fremtidigt arbejde.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfillment of the requirements for acquiring the Master of Science degree in Computer Science.

Lyngby, December 2011

Maciej Krajowski-Kukiel

Acknowledgments

I would like to thank my Supervisors - Michael Kai Petersen and Sune Lehmann - for their support, involvement and mentoring me through the whole project.
I would like also to thank Jakob Eg Larsen for his useful remarks.
I would also like to thank Anders Højbjerg for his support during my whole research.

Contents

Chapter 1

Introduction

We live in times where a lot of resources are invested into serving selective, user-relevant content. This approach called personalization is becoming ubiquitous. The most common example is Amazon's recommendation system: "Users who bought this product also bought this...". It is not only the company who takes benefit of such features by increasing sales. The user is also a winner due to saving time browsing for products. Last.fm, a portal where you can listen to music, went even further and developed a mechanism to measure users taste in music. As a result, the service is able to suggest new bands or songs that users are likely to enjoy. Taking a closer look at these two approaches, one realizes that in fact they are very much the same. First they define a metric to measure similarity between subjects (either users or items) and then they present conclusions . Such concepts can be applied also to any sport or computer game. What is most exciting for players is to challenge an opponent similar to themselves. If the opponents are too weak or too strong, then the match is likely to be one sided, and both sides would consider the time wasted. Hence, creating matchmaking based on user similarity will greatly increase user experience. This thesis provides a proposition of calculating one possible model, which can be used to define similarity between bridge players by estimating their performance. Besides, in the thesis graphical methods have been used to visualize some of the results of applying the approach to real statistics from bridge games.

1.1 Goals and Challenges

There are three main challenges that the thesis is facing. The first one is modeling the bridge player's performance. This will be used as a metric that defines players similarity. Calculating, or rather estimating, it belongs to the family of pairwise comparison problems and has been treated as such. Rather than trying to invent the solution from scratch, the decision has been made to use one of previously researched methods and adjust it to bridge case. Due to the complexity of bridge rules and scoring, this task is challenging and is not expected to end up with the perfect working solution.
The second goal is to apply the model to the real life statistics and analyze the results. Such approach allows to verify the correctness of the developed model. In addition, it exposes its weak and strong sides, making the improvement process more effective and more reliable. There are three main challenges in this domain. One is to acquire a data set which is large enough to be representative. The second one, is to filter and clean the data to reduce noise that might have occurred during the gathering process. The third challenge is to implement the solution and process the filtered data.
Finally, the results of applying the model to the real statistics has to be visualized. It should help to expose the most important features that were mentioned in the previous paragraph, such as the correctness of the model along with its advantages and disadvantages. It can also be used to prove that the assumptions made during the modeling phase were satisfied, or to realize that they were wrong. In either case, the visualization most probably will show what should be the next step in order to obtain better results. Challenges in this field mostly concerns using adequate statistic measurements, analyzing the right parameters, drawing the right conclusions and finally showing the results in a readable manner.

To summarize, the goals of the Thesis can be listed as follows:

- Model bridge player's performance
- Acquire real life bridge statistics
- Visualize the results

1.2 Potential Application

Bridge has been chosen as a point of interest mainly because of willingness to apply players performance as the most important metric in matchmaking al-

gorithm[1]. The current software that is used by most bridge players, including many world champions, allows only to either create a complete custom match or to wait for random players. Besides, the great majority of players have the urge of being able to compare themselves to other players. Having a working model of predicting the skill level of a bridge player is the first and key factor to create such algorithm.
The second potential usage is in rating players by various bridge Federations, take for instance American Contract Bridge League or Polish Bridge Federations. Their current systems seem old fashioned and very simple compared to other popular logical games like Chess or Go. The current way for official measuring of the performance of a bridge player is to reward him after achieving a certain place in an event, where the rank of the event matters. In its current form it is only possible to earn points. It means that the player is not penalized for a bad score, implying that a weaker player just needs to play more in order to obtain a good rating. This results in unreliability of the currently used rating system.
Finally, the model invented in this thesis might be applied to a completely different domain - artificial intelligence. Bridge is a very interesting case of machine learning algorithms, because of its partial observability (players are not knowledgeable about the entire game - they can see only a limited number of cards) and required cooperation between two agents. The obtained model might be a good benchmark for measuring robots effectiveness, and with additional debugging output, it is likely to detect fields that need improvement.

1.3 The Thesis Flow

The Thesis starts with a description of the various approaches that have been developed before in order to model the performance of a player in other sports. No similar work developed for bridge is known to the author. Chapter 3 contains basic information about bridge rules and shows what the scoring in this game looks like. It is crucial to have an understanding of how the results are compared to each other, because they are the core reason of some of the most important decisions taken during modeling. As a last part of this Chapter, there is a list of requirements for the model and each approach described before is evaluated based on it. Chapter 4 describes the approach that has been chosen and the way it has been extended and applied to the bridge case. Chapter 5 uses visualization techniques to present the results of applying the model and verify its strong and weak sides. The last, Chapter 6, summarizes the model and visual results, defines fields of improvement for future work and general conclusions that were

[1] Matchmaking algorithm is beyond the scope of this Thesis and is treated as a hypothetical application possibility, not as a goal to achieve

drawn.

Chapter 2

Previous Approaches of Modelling Player's Performance

Modeling player's performance falls into the problem category of pairwise comparison. It is well known and has been studied by many researchers. Therefore, some of the works on topics related to data analysis (Janert, 2010), pairwise comparison (Chebotarev and Shamis, 2006), rating building (Glickman, 1995, 2001; Ralf Herbrich and Graepel, 2007; Weng and Lin, 2011; Smith, 2006) and network-based solutions for measuring performance (Park and Newman, 2005; James Piette and Anand, 2011) have been analyzed in order to gain insight into already invented techniques. Instead of reinventing the wheel, author decided to build an approach "standing on the shoulders of giants" and choose one of the existing models as a starting point. This Chapter has been divided into sections, each representing one method for calculating players performance. The first one is the Elo rating system - the oldest but also the most popular one. The next two are its successors - Glicko and TrueSkill. All of them are based on Gaussian (optionally Logistic) distribution and they assume that players performance is a random variable that follows mentioned distribution. The next two are unrelated with them and they have been based on network theory. One of them has been prepared for rating the US College Football League, while the second one measures performance of basketball players.

2.1 Elo Rating System

The original Elo method was designed for measuring players performance in Chess. It was created by the chess player Arpard Elo for the United States Chess Federation to replace the Harkness system[1], which was considered too simple. Elo's model defines performance as a random variable following Normal Distribution. Players skill level is built around the fixed norm (for example 1500 points), and then adjusted along with players activity to better represent players actual performance. A key part of the model is calculating the expected score of a game between two players - probability that player 1 outperforms player 2. After the match, the corrections are made to make the real score more likely to happen in the future. It is done through taking away some points from the loser and adding some to the winner. Elo's model in its simplest form is expressed with equation (Glickman, 1995):

$$R_n = R_{n-1} + K * (S - E) \tag{2.1}$$

It is interpreted as follows: Current rating R_n is defined by adding to the previous rating R_{n-1} a difference between obtained score S and expected score E, multiplied by importance factor K. S is either 1, if the player wins the game or 0 if he looses. The draw is considered half a win and half a defeat, which is represented by $S = 0.5$. K is a subjective element and it defines the weight of the game. One could also interpret it as how much trust is given to the previous rating - the bigger K, the bigger change is possible, which means that trust is lower. The implementation of the K-factor is very different from model to model - some systems use a K-factor based on player rating and lowering it if it exceeds a certain value, while others take into consideration only the number of games played by a player. The strength of the first approach is that if the rating is high enough, it is reasonable to increase its trust, basing on the assumption that very good players' progress is much slower (if any) compared to weak players. The justification for the second one is that the more games were played, the more knowledge the system possesses about the performance of players. Both methodologies seem to be valid according to the associations who use them, for example FIDE (InternationalChessFederation, 2011) for the first case and USCF for the second one. (Miller, 2003).

To calculate the expected score Elo suggested to use Normal Distribution, however letter studies on this topic showed that a Logistic Curve is also an option (Glickman, 1995). Glickman, the author of the improvement to the Elo system described in the next Section, claims that in practice it does not really matter which one will be used, however it is convenient to use the letter one (Glickman,

[1] Information about the Harkness System can be found in the book from 1967 called "Official Chess Handbook" written by Kenneth Harkness.

1995). Formulas based on Logistic Distribution to calculate expected scores for Player A and Player B are (Glickman and Jones, 1999):
Player A:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}} \tag{2.2}$$

Player B:

$$E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}} \tag{2.3}$$

The important observation is that the result is different for each player that agrees with the common sense. A player with the higher rating should be more probable to win than his lower ranked opponent (or the other way around - the low ranked player should not be probable to win versus the higher ranked opponent).
The Elo rating system is the most popular rating system in the world. It is used for example by FIDE (World Chess Federation) (InternationalChessFederation, 2011), World Football rating (Runyan, 1997), European Go Federation (EuropeanGoFederation, 2011), Major League Baseball and American College Football. Moreover, it is also a very popular rating system used in On-Line games. It has been implemented in the most popular of Blizzard's games, such as World of Warcraft or StarCraft II. In addition, it is used by the recent, very successful game called League of Legions. Yahoo! Games has also adopted this way to build players rating. The method, because of its age[2] is very well understood and analyzed. This is a very big advantage of an established system, which should not be omitted. There are a number of mathematical and statistical explanations of why this system works. However, there are some well known issues with the system as well. Probably the biggest one is that it is designed only for two players games. This is a serious limitation, since many games (including bridge) involves more than two players. However, many workarounds for this issue have been developed, which make it possible to use the Elo system even for team games.

2.2 Glicko Rating System

The Glicko rating System is an improvement to Elo's solution, created by Mark Glickman. Glickman tackled the problem of the reliability of players rating. When two players with the same rating face each other they both have a probability of winning of 0.5. The problem in this assumption, is that one of them could not play for a a very long time while the second one could play regularly

[2]Elo invented his system in 1978.

(Glickman, 2001).
He introduced another parameter that needs to be estimated - the rating uncertainty RD. It stands for rating Deviation and is nothing more than a well known standard deviation (σ). For his model Glickman suggests using 350 as startup value for RD. The formula (Glickman, 2001) to recalculate it after a certain, fixed period is:

$$RD = min(\sqrt{RD_{old}^2 + c^2 t}, 350) \tag{2.4}$$

The t is the number of the rating periods between the current one and the last activity of a player. If the player has not omitted any rating periods (i.e. he had played in the previous one) then $t = 1$. The second variable, c is a constant, and describes how much the uncertainty should increase for each rating period during which the player has not participated. There are two ways described by Glickman to derive a proper value of c. The first one - usually an expensive one - is to analyze the data. The second one is to define how many rating periods must pass in order to make the rating of the player as unreliable as a rating of the new player. The example assumes that if the typical RD for a player is 50 , the rating period is two months, and 5 years are needed in order to completely distrust the rating (which means, that $t = 30$ ($30 * 2$ months is 60 months, which is 5 years), the equation (Glickman, 2001) could be written as follows:

$$350 = \sqrt{50^2 + c^2 * 30} \tag{2.5}$$

Hence, for this case $c = 63, 2$.
Equation 2.4 ensures, that no matter how long a break the player would have, his uncertainty will never be greater than the uncertainty of a new player - it is due to the min function.
After adjusting the RD for each player that participated in the rating period, the next step is to calculate the actual rating and the new RD. In order to calculate the new rating r' for each player p_j facing m number of opponents, which ratings are r_1, r_2, ..., r_m and RDs are RD_1, RD_2, ..., RD_m and output of the game versus them is respectively s_1, s_2, ... , s_m which can be either win $s = 1$, draw $s = 0.5$ or lose $s = 0$, the following equation is used (Glickman, 2001):

$$r' = r + \frac{q}{1/RD^2 + 1/d^2} \sum_{j=1}^{m} g(RD_j)(s_j - E(s|r, r_j, RD_j)) \tag{2.6}$$

$$RD' = \sqrt{(\frac{1}{RD^2} + \frac{1}{d^2})^{-1}} \tag{2.7}$$

The exact formulas of d^2 and $G(RD_j)$ are not relevant for explaining the concept and the way of calculating them can be found in (Glickman, 2001). What

is important, is that they are based on opponents RD' and ratings. One should note, that there no longer exists an artificial K factor - instead the systems is using these two variables to determine the new rating.
It might be confusing at first glance why RD is calculated twice. The explanation is that the first step expressed in Equation 2.4 is focused on increasing uncertainty depending on how many rating periods were omitted. The second one is responsible for lowering it, based on opponents skill level and uncertainty of their performance.

The calculation of the expected game output (Equation 2.8) is very like Elo's way of calculating it (Equation 2.2). The only change is taking into consideration the uncertainty of the opponent's rating (Glickman, 2001):

$$E(s|r, r_j, RD_j) = \frac{1}{1 + 10^{-g(RD_j)(r-r_j)/400}} \tag{2.8}$$

2.3 TrueSkill

Glicko, similar to Elo, has been designed for two players games. However, TrueSkill, which has been developed by Microsoft and is based on Glicko does not have this limitation. It allows not only for an unlimited number of players to participate, but also makes it possible to derive the skill level for each individual basing on the result of his team. Moreover, it explicitly models the probability of a draw, instead of treating it half a win and half a lose.
TrueSkill is using Bayesian Statistics, which means that posterior belief is a result of obtaining new evidence to the prior belief. The main difference between Bayesian statistics and the classic (frequentist) approach is that no additional assumption of repeating observations many times must be added (Janert, 2010). In addition, the model uses factor graphs (Kschischang et al., 2001) for an efficient inference process Ralf Herbrich and Graepel (2007). The process of defining an updated skill level for a player after a match begins by acquiring prior information about his skill level s_i, which is described by a normal distribution with most probable performance μ and its uncertainty σ. The higher the doubt in someone's rating, the bigger the change to his skill after a game is possible. To prevent the standard deviation of always going lower - meaning almost no change is possible after playing several of games - it is artificially increased by the dynamic factor τ. Hence, the prior information could be written as (Ralf Herbrich and Graepel, 2007):

$$\mathcal{N}(s_i : \mu_i, {\sigma_i}^2 + \tau) \tag{2.9}$$

In the next step a β factor is used to model players performance also with Normal Distribution. The new factor can be interpreted as a number of points by which players must differ in order to say that odds of winning are like 80% to 20%. It is a fixed value, which depends on a game. The less random the game is, the lower β will be. However, if a game includes a lot of randomness - for example poker - the higher it becomes. Player's performance is written as (Ralf Herbrich and Graepel, 2007):

$$\mathcal{N}(p_i : s_i, {\beta_i}^2). \tag{2.10}$$

After this, a performance of a team t_i can be calculated. It is a basic sum of all performances of all team members p_i, with additional weight w_i (Ralf Herbrich and Graepel, 2007):

$$t_i = \sum_{i=0}^{N} w_i s_i. \tag{2.11}$$

The weight is added to increase fairness. The default value is 1, which means that a player played 100% amount of time in a game. However, one can easily think of a situation when someone is being disconnected from the game.
Once the performances of all teams are known, it is possible to compare them, which is done using TrueSkill to count the difference in performance d_i between teams. Based on the result, one can decide whether one team has won - meaning that $d_i > \epsilon$, lost - $d_i < \epsilon$ or there was a draw - $d_i = \epsilon$, where ϵ is predefine number representing a threshold for a draw. For TrueSkill, it is only important who won the match - it ignores information about by how much. Once all results are calculated and taken into consideration, which requires a few iterations if many teams are involved, one could update a rating and uncertainty about each individual. The idea is relatively simple: the more likely the outcome was, the lower change to μ and σ. On the other hand, if the result was not likely to happen, then the change to μ is much greater, but the uncertainty is also lowered, since there is new important evidence. The detailed math about updating ratings and uncertainty can be found in Ralf Herbrich and Graepel (2007); Moser (2011).

Recently, another Bayesian approach for on-line rating (Weng and Lin, 2011) has appeared. It is also based on Glicko, and gives comparable approximations as TrueSkill, however it is superior in efficiency for a multiple team game scenario to a system created by Microsoft. It is because it does not require iterative calculations in such case. All main ideas however remain the same.

2.4 Football Rating for US College League

A completely different approach has been presented in the solution for building rating for a US College Football Team. The problem they had to tackle was to define the best College team in the USA. This would not be challenging if the usual procedure was applicable: Creating a tournament for X best teams and let the winner claim the title. Instead, the result had to be based on the results during the season. What made the task specifically difficult, was that college teams are divided into conferences, and about 75% matches were played within their area. As a result, many teams have never played against each other.
The previous system contained a big subjective factor, being the notes of coaches and sport journalists. This made a lot of people feel that the process of choosing the best team was unjust. The new approach has been based on network theory. Here each node is defined as a team and each directed edge from node A to node B as a representation of team's A win over team B. The problem of the uneven strength distribution between conferences has been solved by additional assumption, similar to the one made in collaborative filtering. Indirect wins have been introduced, meaning that if Team A has won versus Team B, and Team B won versus team C, then if Team A has not played vs Team C, it is still considered that they are better than team C. They did not limit themselves to 1-level indirection, instead they have added an α factor, which power corresponds to indirection level, meaning the higher indirection level, the lower influence of such win. Their representation of win-lose scores is a matrix, where element A_ij is the number of wins of team j over team i (usually 0 and 1, but sometimes 2). The equation to calculate which team is the best is rather simple (Park and Newman, 2005):

$$s_i = w_i - l_i \tag{2.12}$$

Where w_i is score from winning and l_i is penalty for loosing. They are given with formulas (Park and Newman, 2005):

$$w_i = k_i^{out} + \alpha \sum_j A_{ij}^T w_j \tag{2.13}$$

$$l_i = k_i^{in} + \alpha \sum_j A_{ij} l_j \tag{2.14}$$

The first element in equation (k_i^{out} and k_iin) is the number of direct wins and loses, while the second one defines indirect wins and loses. This approach provides a powerful way of dealing with problems, which are related to lack of information. In addition it takes into account the strength of opponents. The stronger the beaten team is, the higher a score will be gained. Simultaneously, loosing versus a weak team costs a lot more than versus the strong one. The whole approach is described in article (Park and Newman, 2005).

2.5 Basketball Player's Performance

The goal of this approach is to evaluate how important each basketball player was for each team he played in, compared to his teammates. Moreover, it is supposed to give an answer to the question of how well he performed statistically. In order to answer this, authors have, in two steps, built a network which allows to generate two measures.
They start with constructing a unimodal graph, where a node is a player and weighted edge between two players exists if they have played at least once for the same team. The weight represents the strength of their interdependency, taking into account the team's performance. The better the team played, the higher a value is assigned. In order to measure the team's effectiveness, they analyze statistics containing the number of points scored while in offense or allowed to score while defending. The final unimodal graph is derived through a bipartite graph, where there are two types of nodes: Player and Unit. From the definition of bipartite graph, no edge between player-player or unit-unit may exist. This bipartite graph is represented by incident matrix W - a row represents a unit, while a column a player. The value in cell i,j is referred to as w_{ij} and it shows how well the unit has performed. The final unimodal graph is by calculating $A = W^T W$ (James Piette and Anand, 2011). They use eigenvector centrality with random restart. Then, they used the concept of Latent Pathway Identification Analysis (Pham et al., 2011), which has been originally used in biological networks. The idea is to generate random walks in order to assess the importance of each player relative to all the others, which exist in the network. The result of this process is a centrality score, where the statistical significance is calculated by using a so-called bootstrap technique (Janert, 2010; James Piette and Anand, 2011). They distinguish three different statistics: offensive performance, defensive performance and total performance. The process described above is used to calculate each of them, meaning that the final output are three unimodal weighted graphs - one for each metric. The whole approach is described in details in (James Piette and Anand, 2011).

Chapter 3

Choosing the Approach

Alongside the necessary background knowledge gained in Chapter 2, more input is needed to develop a reliable method of tackling the problems introduced in Chapter 1. One of prerequisites is establishing an understanding of how the score is measured in bridge. This topic is covered by the first and the second section of this Chapter. Grouping all the approaches described in Chapter 2, and choosing the one that best fits bridge is covered by the last section. In addition, it introduces a list of requirements for the model to make a meaningful comparison between different choices possible.

3.1 Scoring in Bridge

Bridge is a card game for exactly two partnerships (4 players) at one table. However, unlike other sports, take for instance chess, football or basketball, in bridge, the score of a player is not solely based on the grounds of ones own result. Instead, the same card distribution is duplicated and at least another 4 players play the same deal as the first table, and then the scores are being compared [1]. This Section will explain what it means.

[1] There is one very common variation of bridge scoring called rubber bridge, in which there is no comparison taking place. However, in general there is at least one other table, to which

The description starts by showing the flow of playing a deal and the basic terms involved. A *deal* is a certain concrete distribution of 52 cards between four players - each player receives 13 cards. These 13 cards are referred as a *hand*. Each of the players sit at one of four positions - *North*, *South*, *East* or *West*. In fact, one might well establish, that the hand belongs to the position, not to the player. A set of 4 players is called *table* and a match between them is referred as a *game*. One deal must be played by at least two independent tables. Players sitting opposite each other are partners and their goal is to obtain the highest possible amount of points. Hence, by saying that in one *deal* N tables are participating, one thinks about exactly $N*4$ players and $N*2$ pairs. Partnerships are commonly referred to as *line* at which they sit: *North-South* or just *NS* and *East-West* or just *EW*. It is of significance which hand belongs to which position. If the cards between position N and S were swapped, it would be considered a completely different deal. To calculate the total number of deals that can be produced the following equation must be solved:

$$\binom{52}{13} * \binom{39}{13} * \binom{26}{13} * 4! = 1,287,473,706,371,731,028,141,698,560,000 \quad (3.1)$$

This is an extremely large number. Hence making the assumption that each deal is unique is very reasonable, since it is not probable, even in the long run, that the same deal is randomized twice.

After the distribution of the cards, the actual play begins. The person who dealt the cards becomes a *dealer* and he starts the *Auction*. It is not necessary to describe this part of the game in details, only to mention that the partnership who bids the highest *contract* has won the auction, and one of the players becomes a *declarer* and his partner becomes a *dummy*. The dummy is not participating in the deal anymore. Each of their opponents will be now a *defender*. The observation can be made that the dummy participates only in the first phase of the game, while the declarer plays alone against two players. The declarer's goal is to make his contract, which means he tries to take at least as many *tricks*, as he has claimed to during the auction, while the defenders do everything in their power to avoid this. A trick is when each of four players play a card. The trick belongs to the person who has played the highest card in the suit, or the highest trump colour.

Depending on the final number of tricks, if the declarer has made his contract, he and his partner get a certain number of plus points (while defenders get the same amount of minus points). The amount of points depends on the contract, the number of tricks taken and the vulnerability[2]. If the declarer fails to take

you compare your result. Only this scenario is taken into account in this thesis - it is called "duplicated bridge".

[2]There are two possible vulnerabilities - vulnerable, commonly described as *red* and not

the required number of tricks, the defenders get a plus score, depending on how many of the tricks declared were missed and whether the contract was doubled or redoubled, while he and the dummy get the same amount of minus points. Knowing the exact rules of giving points is not relevant to the solution, hence it will be omitted. What is however important is the way of calculating the final *score*, and this is the subject of the next section.

3.2 Comparing Results in Bridge

It is not hard to get many points by a partnership who posses all good cards. Thus they should not be awarded for just being lucky and getting a lot of Aces and Kings. Rather it is their intellectual effort, which should give them a good score. Duplicate bridge tries to minimize the randomness of cards distribution by duplicating a hand to multiple tables and comparing the results. The effect is that the term "opponents" refers not only to the second pair at the table. All players at the same line are challenged simultaneously as well. Achieving the best score means more than just beating a pair who is faced directly. In fact it is outperforming (indirectly) all other pairs holding the same cards. It is important to notice that players at other tables who sit at opposite lines are actually teammates, because they prevent opponents in achieving a good score. This observation is an important one, and should be the used in the final approach.

3.2.1 IMPs and Matchpoints

There are many ways in which partnerships' scores can be referred to each other, however two of them are most commonly used in bridge. The first one is called Matchpoints. A partnership gets 2 matchpoints for each score that is lower than their, 1 matchpoint for each tie and 0 otherwise[3]. Hence, if a partnership beats all other pairs, it gets 100% (commonly referred to as top). The worst pair receives 0% (which is known as bottom). This type of scoring generally encourages an aggressive style of playing, since it does not matter how much the result differs from the others; it is only important how many it outscores or

vulnerable, commonly described as *green*. There are hence four possible combinations - either both are red, both are green or one of the sides is red and the second one is green. Vulnerability simply means, that you can get more points if you win the auction and make your contract, but you are risking a higher amount of minus points if you win the auction and fail to collect the required number of tricks.

[3]In the USA instead of 2,1 and 0, partnership obtains respectively 1, 0.5 and 0.

0 - 10 = 0	220 - 260 = 6	600 - 740 = 12	1750 - 1990 = 18
20 - 40 = 1	270 - 310 = 7	750 - 890 = 13	2000 - 2240 = 19
50 - 80 = 2	320 - 360 = 8	900 - 1090 = 14	2250 - 2490 = 20
90 - 120 = 3	370 - 420 = 9	1100 - 1290 = 15	2500 - 2990 = 21
130 - 160 = 4	430 - 490 = 10	1300 - 1490 = 16	3000 - 3490 = 22
170 - 210 = 5	500 - 590 = 11	1500 - 1740 = 17	3500 - 3990 = 23
			4000 & over = 24

Figure 3.1: Table to look up the number of IMPs for given difference in points range.

ties. Hence, even a small difference in the number of points, for example 120 instead of 110, may bring a lot of matchpoints.

By contrast, the second method called International Match Points (IMP) generally favours a more 'safe' style of playing. A pair gets points based on how much it differs compared to other pairs. The lookup table is presented in Figure 3.1. The calculation is quite simple when only two tables are involved. The result of N-S from table 1 is deducted from the score in table 2, and a proper score is assigned based on the lookup table. However, if there are more pairs in play, then there are more choices. For example, the average result could be calculated (summing all results and dividing the sum by the number of tables) or one could use the median instead. Some algorithms discard a few top and bottom results from the calculations. Another approach is to calculate imp difference for each possible pair of results (if there are 5 tables, then the result from table 1 is compared to table 2, 3, 4 and 5, then result from table 2 is compared to the one achieved in tables 3, 4 and 5 etc.) and thus getting an average.

In duplicate bridge, events (for example tournaments, championships etc.) might be played in three categories: individual, pairs and teams. In the first two, all players results are compared to each other. The difference between them is, that in the first case partnerships change in every round (which is usually 2-3 deals) and in the second one, the partnerships are constant. Any type of scoring is appropriate to use - matchpoint and IMP, though the first one is probably more popular. Things change in case of a team event, which is considered the most prestigious. During such an event, each team is represented by 2 partnerships, hence there are two tables. Pairs from the same team must play at different tables at different lines. The results from each table are later compared to each other (based on the table showed in Figure 3.1 and the winner is the team with the higher score. Even though matchpoints could be a viable approach, it is too imprecise. The only possible scores in case of two tables are: 100%, 50% or 0%. It means that difference of 10 points in a result is equally important as a 2000

points difference, what is unjust. Hence, IMP is used during all important (and usually also unimportant) team events.

3.2.2 Average Cross-IMPs Algorithm

Because IMPs are more reliable and might be applied to any event type, it seems natural to use IMPs scoring in the thesis. There are many methods that can transform a set of results in points into IMPs. The one that will be used in the thesis is called average cross-imps. The example below explains how it works:
Assume there are 5 tables that are playing the same deal - all North players have the same cards, all South players have the same cards etc.
The results from table 1 to table 5 are as follows (all points are relative to NS):

-100, 600, -50, 620, -100

In order to get imps for table 2 (which got 600 points), its result is compared to all of the others:

to table 1: 600 - (-100) = 700
to table 3: 600 - (-50) = 650
to table 4: 600 - 620 = -20
to table 5: 600 - (-100) = 700

To change the difference in points into imps, one should do a proper lookup in the table showed in Figure 3.1. In this case, the proper values are as follows:

Versus table 1: 700 = 12 imps
Versus table 3: 650 = 12 imps
Versus table 4: -20 = -1 imp
Versus table 5: 700 = 12 imps

The final score of this deal for NS from Table 2 is the average of imps calculated before. Hence, it is:

$$(12 + 12 + 12 - 1)/4 = 35/4 = 8,75 \tag{3.2}$$

The score for EW can be obtain by multiplying the result for NS by by -1, which is -8,75 imps.

There is an important thing that should be pointed out before finishing this chapter. The NS pair from table 2 might have got a relatively high final score

of 8,75 imps not only because of their great play or their opponents mistake. It could happen because the other pairs sitting at *their line* - NS - have done a relatively poor job *or* the pairs that were sitting at opposite lines - EW - have done a relatively *good* job. It is important to note, that actually NS from table 2 would like all other pairs sitting on NS to be as weak as possible, while all EW pairs, except their direct opponents, as strong as possible.

3.3 Possibilities of Applying Existing Approaches to Bridge

After getting familiar with the basic bridge rules it should now be possible to get an overview of the problems and traps that have to be avoided. The range of requirements, that should be satisfied by the model of player performance is listed below:

1. Distinguish between good and bad opponents
2. Distinguish between good and bad partner
3. Take into account how many games partnership played together
4. Take into consideration the skill level of players at other tables
5. It is not only important, who win/lose, but also by how much
6. Consider extreme cases, when world champions win (lose) against beginners

The first one is about awarding/penalizing players differently if they play versus a weak or a strong opponent. Similarly, the skill level of the partner should be taken into consideration to avoid being penalized/awarded too much by someone else's mistakes/correct decisions. In addition, partners understanding of each other is very important. Two regular partners are able to win with much better opponents who play against each other for the first time. The fourth requirement is reducing penalty for "being at the wrong place at the wrong time". It covers those cases, when a weaker opponent at the other table plays against a strong one and they generate unnatural high scores. Requirement 5 has to be satisfied in order to reward players that make a decision that will bring plus score in the long run, even though they could often lose by a very small number of IMPs. Moreover, if a player loses a lot of imps several times, it is a strong indication that his opponent is much better. Requirement 6 indicates that extreme scenarios

should be verified. It is quite common, especially while playing on-line, that situations where beginners play with champions occur and the model should take such cases into consideration.

3.3.1 Analysis of Various Rating Systems

The list of requirements defined above has been used to compare and analyze approaches introduced in Chapter 2. Starting from Elo, the method to determine whether an opponent and partner are high skill players or not is built-in by default, because it is a rating system. However, the limitation of the model being only applicable to two players games complicates a comparison. There are, however, solutions to reduce bridge into a scenario, where two "players" are playing versus each other. One such method is to treat the pair as one player, as suggested in an on-line article (Curley, 2010). The idea is to count every existing pair combination as completely separate players. It means, that if player A plays together with players B and C, the two artificial players are created, representing these two partnerships. Choosing this approach would make it hard to derive the skill level of individuals, which is the final goal. However, what can be done, is to find the average rating of players and use it in the calculations.
There are no difficulties taking into account the number of games played with each partner. An additional parameter might be introduced, which modifies expected score. Requirement 4 make things a little bit more complex, since it also requires to take into consideration all other pairs that played the same deal on other tables. However even for that solutions have been found, for example by the Days of Wonders(Allen and Appelcline, 2006). Their approach is to split one game, in which many different players without teams participated (so called FFA - Free For All) into many duels (player vs player). The winner of the game has won with all other players, the player in the second plays with all except one etc. Hence, it is possible to calculate new ratings for each artificially created duel and take the average as a final result. Actually, the described way does not differ from computing IMPs in the example in the previous Section!
Requirement 5 is relatively easy to solve by editing the K factor. Such approach has been used for example to calculate ratings of World Football (Runyan, 1997).
Elo has no difficulties with extreme examples, which are the subject of Requirement 6. If the difference between players is very big, then one of them gets almost no points at all, while the second one gets a very large number. Intuitively this is what should happen.
To sum up, the Elo rating system is able to satisfy all requirements that have been defined. Moreover, it is worth stressing again that it is well established, deeply analyzed and broadly used on many platforms.
TrueSkill and Glicko are improvements to Elo, hence they automatically satisfy all requirements.

3.3.2 Analysis of Network-Based Approaches

Separate analysis is required for network based approaches. The solution applied to the case of the US College Football League is a very tempting one. There is however the question, how this approach might be applied to bridge to satisfy all requirements. The second concern is whether this solution will be scalable for a huge amount of players. It is due to the main assumption, on which the approach is based, that if A won with B and B won with C then A won with C. One additional deal requires rebuilding a very large part of the graph, depending on how many levels the assumption is to be applied to. One should remember that one deal, in which take for instance 16 tables are involved, requires changing the rating of 64 players, and because of the above assumption, modifications have to be done to all players that have played at least one game with any of them.
As it goes about applying this approach to bridge, one possible graph representation is to define each node as a partnership, and each weighted directed edge between two partnerships would indicate by how much (in IMPs) one partnership won (lost) with the other one. It could also be necessary to add a second weight which would indicate how many deals have been played, because it makes a difference whether a small advantage on one side is due to a few games played between two partnerships or because of very close matches. Then, the performance of each player could be a weighted (by the number of deals played) average of all partnerships to which this player belongs. However, this way of defining graph does not provide a way to take into account the skill level of the partner and all other players involved in the game. It treats the partnership as one unit, meaning it does not matter whether it consists of players with the same skill or not.
Another way of defining the network would be to create a node that represents a deal. It would be created for each deal in the dataset. Next, for each table participating in it, four nodes will be created and they will be connected to each other with an edge, indicating they were playing at the same table. The names of those nodes could be the position, from which a player plays - North, South, East and West. Beside, a node that represents a player should be created, in order to keep track of which player played what deals. Figure 3.2 should demonstrate the idea and how it could look for one hand played by three tables. This interpretation would allow to store all necessary information about partners and opponents of the player in each deal. It will allow to introduce a technique to apply a partner's skill level and to opponents, for example by using theirs w_i and l_i. However, an additional issue occurs - the bootstrap problem. It is not known from which player one should start calculating the score. The size and traversing complexity of the graph will be very large even without taking into account when each deal has been played. Depending on this, the results might vary dramatically. Moreover, it would not be possible to model the progress of

a player - only pure result (score?).

To sum up, the approach used by College Football is very interesting, however it does not really apply to bridge. The rating system seems to be superior in every aspect and no single advantage of the network approach has been found. The same reasoning applies to the last approach - measuring Basketball players performance. It introduces a lot of additional problems, inaccuracy without giving anything in return.

3.3.3 Why Elo

The only concern left is which of the three rating systems: Elo, Glicko or TrueSkill should be used. The advantage of Glicko over Elo is taking into account the player's performance reliability. However, this is not that important in bridge. What really matters is the uncertainty of a combined performance of partners, which should be based on the number of games played together and when they were played. This is due to the fact that measuring trust in an individual rating is not very useful - there is a great difference between players who play everyday but with different partners and players who play seldom but always with the same partner. In case of bridge, the second one playing with his regular partner is much more reliable than the first one.
TrueSkill can be considered as a generalization of the Elo and Glicko systems. It introduces a lot of new features, like team handling, draw probability, Bayesian statistics instead of frequentist. It is probably the one that will be able to give the best predictions. The problem with TrueSkill is that it requires a very good understanding of the previous approaches to be used correctly. Such knowledge can be gained only while working with Elo itself. Going with such powerful, but also complex solution without previous experience with rating systems does not seem to be the correct approach. In the worst case it might lead to completely wrong conclusions. In addition, there is no research known to the author, on applying rating system to bridge, which would compensate lack of insight into rating systems. It is another reason to be rather careful of putting all resources into a technique that is likely, but not guaranteed to work.

After considering all pros and cons, Elo has been chosen to be used. Even though TrueSkill is more likely to give more accurate predictions, it is dangerous to choose this method without any prior knowledge about the problem domain. Such insight might be gained only during work on the solution. That is why it was chosen to first develop a prototype of the solution built over a basic Elo model. Such approach is usually chosen by companies who would like to test a new product without risking too much. Moreover, it would allow not only to become familiar with the underlying statistics and to explore the bridge

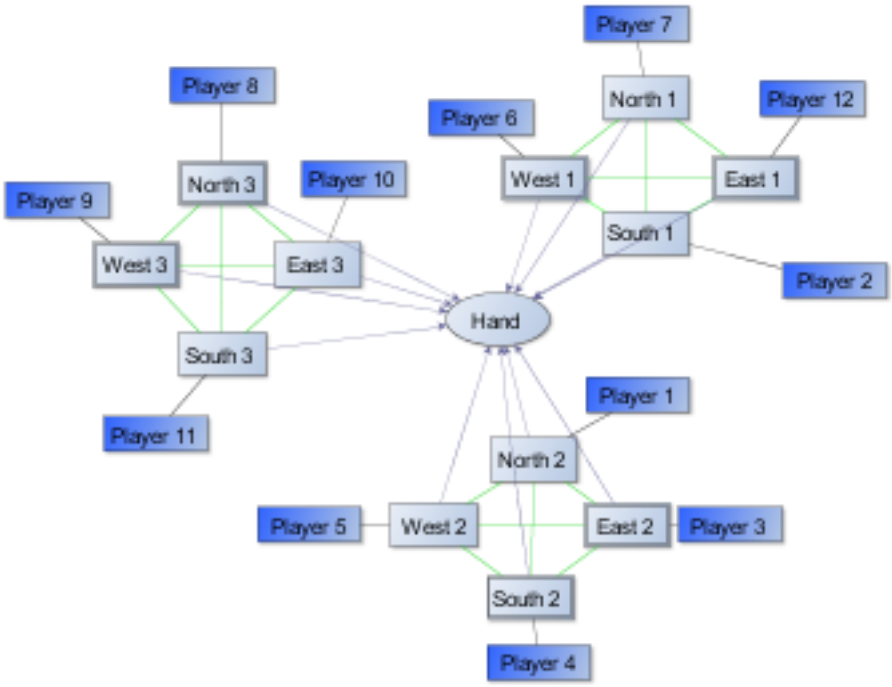

Figure 3.2: The figure contains sample graph. There are 6 node types: 'North', 'South', 'East' and 'West' indicate the position in the particular deal, 'Deal' represents a deal played by many tables and 'Player' is a representation of a player. North, South, East and West are connected if they played at the same table. A player is connected to each position on which he played. He is hence indirectly connected to certain number of deals. Such graph representation has the advantage of storing very detailed information, but the price is an enormous number of nodes.

domain, but it will also be possible to reconsider the list of requirements and set the priorities of what is the most important in bridge. It is important to notice, that TrueSkill does not resolve the most important problem of predicting bridge outcome - namely the partnership history. It also ignores by how much one team won over the second one. It means that a lot of extensions would have to be done to TrueSkill to apply it to the domain of bridge. It should be observed that using a basic method would allow to perform much more experiments, from which it will be possible to draw more precise conclusions. It is because the only parameters in Elo is rating difference and K-factor and there is no overlap between these two. It is also important to note that if for some reason it turns out the rating system is not possible to use for bridge, there would still be enough time to switch to another approach. Finally, the most important argument: nothing can be lost by choosing Elo. If it turned out that the rating system is applicable, however Elo gives poor predictions, all the knowledge and insight gained from the first try of applying it could always be migrated into TrueSkill. Also, developed statistics for measuring accuracy will also be applicable to TrueSkill. Even certain parts of the implementation will be able to be re-used (even though it will be necessary to greatly extend it).

Chapter 4

Modeling the performance

The input from the previous chapter is crucial for building a proper model. From many reasons it was decided to use the Elo rating system (See Chapter 3) as a starting point. Hence, the start-up formula is:

$$R_n = R_{n-1} + K * (S - E) \tag{4.1}$$

The description is divided into two sections. The first one discusses one of two parameters - E - the expected score of a game. The second section describes in details the K factor.

One of the most important things that has to be decided regardless of these two parameters is how to define a winner. The simplest and most intuitive approach is to let the pair who got a plus score become the winner. However, in such case the draw will almost never happen if many tables participate in a deal. The solution is to provide a 1 IMP margin, which would protect pairs from outliers without harming the real winners. Hence, there is a winner if one of the pairs gained more than 1 IMP, otherwise the result of the game is considered a draw.

Another very important decision to make is how often ratings will be updated. The excessive amount of time has been spent to be able to calculate ratings after every game, so the player will not have to wait for an update. A number of experiments were performed on the whole filtered data set (See Section 5.1)

to measure the model's effectiveness, enhancing predictions of expected score and proper weighting by the K-factor. The metric used initially to compare the model's results was Root Mean Square Error (RMSE) and it has been used in the first competition called "Elo versus the Rest of the World" (Sismanis, 2010):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i}^{N}(\hat{\theta}_i - \theta_i)^2} \tag{4.2}$$

The N is the total number of hands played at all tables for $10,000$ deals. The idea was that the higher number of iteration, the lower the average total error should be. Unfortunately, even though one could determine which coefficients work better than the others, the model did not seem to behave correctly. The more games played, the bigger the error was. There were no exceptions. The conclusion has been drawn that it should not be possible to give reliable results with a rating period of 1 game. The justification seems to be intuitive: in one game anything can happen and adjusting the score based on one game will be very prone to noise. In the end, the rating period has been extended to 3 days. It is because the data that have been gathered contain statistics from 21 Feb to 13 May, which is 82 days, which makes 27 rating periods. It has been taken into account that some periods have to be considered as a training set - otherwise there will be very large noise at the beginning and the final score will be polluted.

Having chosen a rating period of 3 days and with a clear definition of winner and loser, one can proceed to the description of two components of the Elo system: rating difference and K-factor.

4.1 Rating difference

To repeat, the most important assumption made by Elo was that it is possible to build a players ratings distribution around the norm. Further research showed that instead of Normal Distribution, it is also possible to use Logistic Distribution, however according to Glickman (Glickman, 1995) it does not really matter which one will be used:

> "For practical purposes, the two curves are indistinguishable."

The shapes of these two are a little bit different, as shown in Figure 4.1, however the whole approach remains the same. The assumption made by Elo is used to calculate the probability that player P_A outperforms (in the long run) P_B.

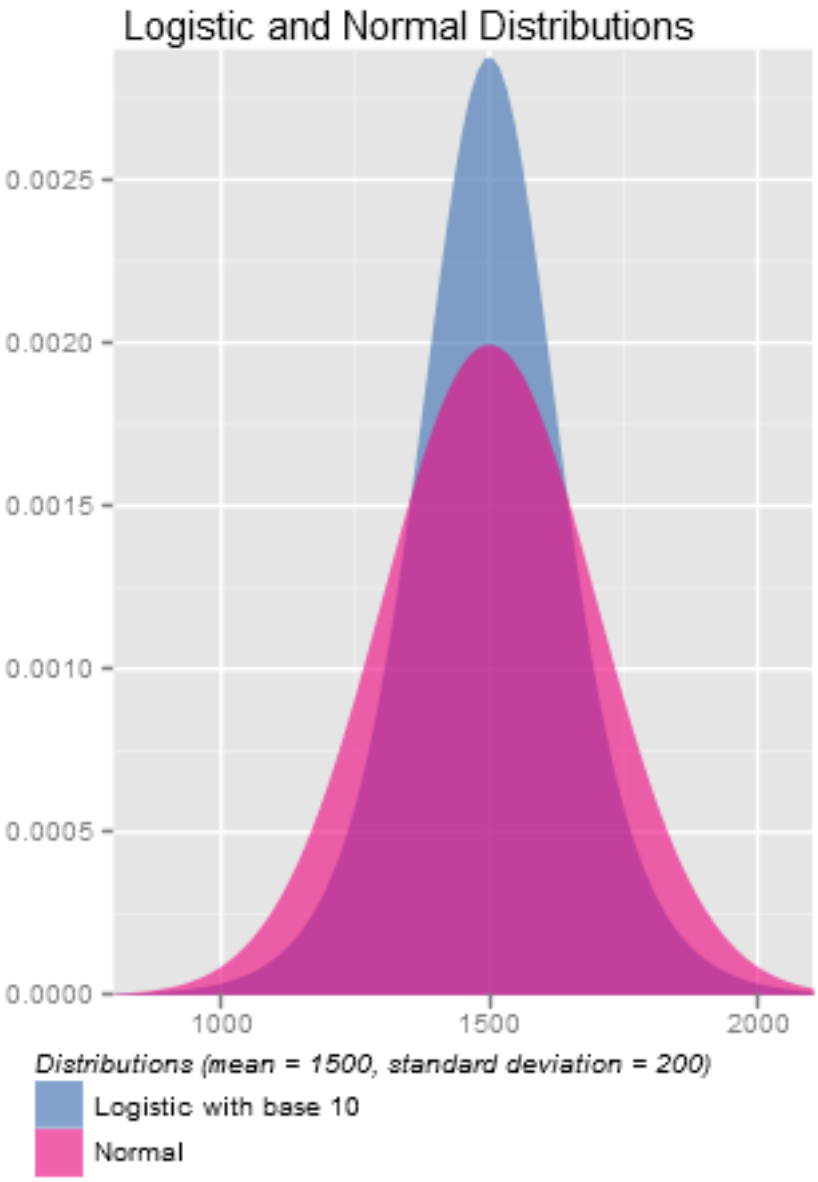

Figure 4.1: The Figure shows the difference between two distributions - Normal and Logistic (with base 10).

However, so far it was not explained exactly how predicting scores works. This is done in subsection 4.1.1. After describing the statistics behind predicting the output of the game, the following subsections describe the improvement of the basic Elo formula, introducing new parameters that are meant to enhance predictions.

4.1.1 Predicting Score

The formula I have chosen to predict the output of the bridge game is the one suggested by Glickman and which is used in plenty of other systems (Glickman, 1995):

$$E = \frac{1}{1 + 10^{rd/400}} \tag{4.3}$$

The 400 is a standard deviation σ (or s) and rd is rating difference, which is calculated as $r_{\text{opponent}} - r_{\text{player}}$. could be set to any other value, some examples of such and their corresponding winning probabilities are presented in Figure 4.2. The reason why 400 is being used is a matter of convention and does not influence on accuracy. Because of choosing this exact value, the player who is ranked 200 points higher than his opponent is considered as one class better and his probability of winning is 76%, while that of loosing, only 24%. This can be verified by solving the equations:

$$E_A = \frac{1}{1 + 10^{-200/400}} = \frac{1}{1 + 10^{-1/2}} = \frac{1}{1 + 1/\sqrt{10}} = 0.76 \tag{4.4}$$

$$E_B = \frac{1}{1 + 10^{200/400}} = \frac{1}{1 + 10^{1/2}} = \frac{1}{1 + \sqrt{10}} = 0.24 \tag{4.5}$$

It might be noticed that the value of 200 is corresponding to the σ of players rating distribution and that is why the value of 200 is considered a different class. One can derive this conclusion after realizing that Equation 4.3 is corresponding to the logistic distribution obtained from subtracting two logistic distributions with different μ (player's rating) and the same σ (fixed standard deviation). According to the definition, the output distribution is given with $\mu_o = \mu_1 - \mu_2$ (which in the Equation 4.3 is described as rd - rating difference) and $\sigma_o = 2\sigma$ (what in the Equation 4.3 is 400). Since $2\sigma = 400$, then each of players rating distribution has standard deviation $\sigma = 200$.
To illustrate this graphically, a Figure 4.3 has been created. It represents a match between two players, one with $\mu = 1300$ and the second one with $\mu = 1500$. As explained above, both have the same fixed standard deviation $\sigma = 200$. Player 1 wins when his performance (showed in x axis) will be greater

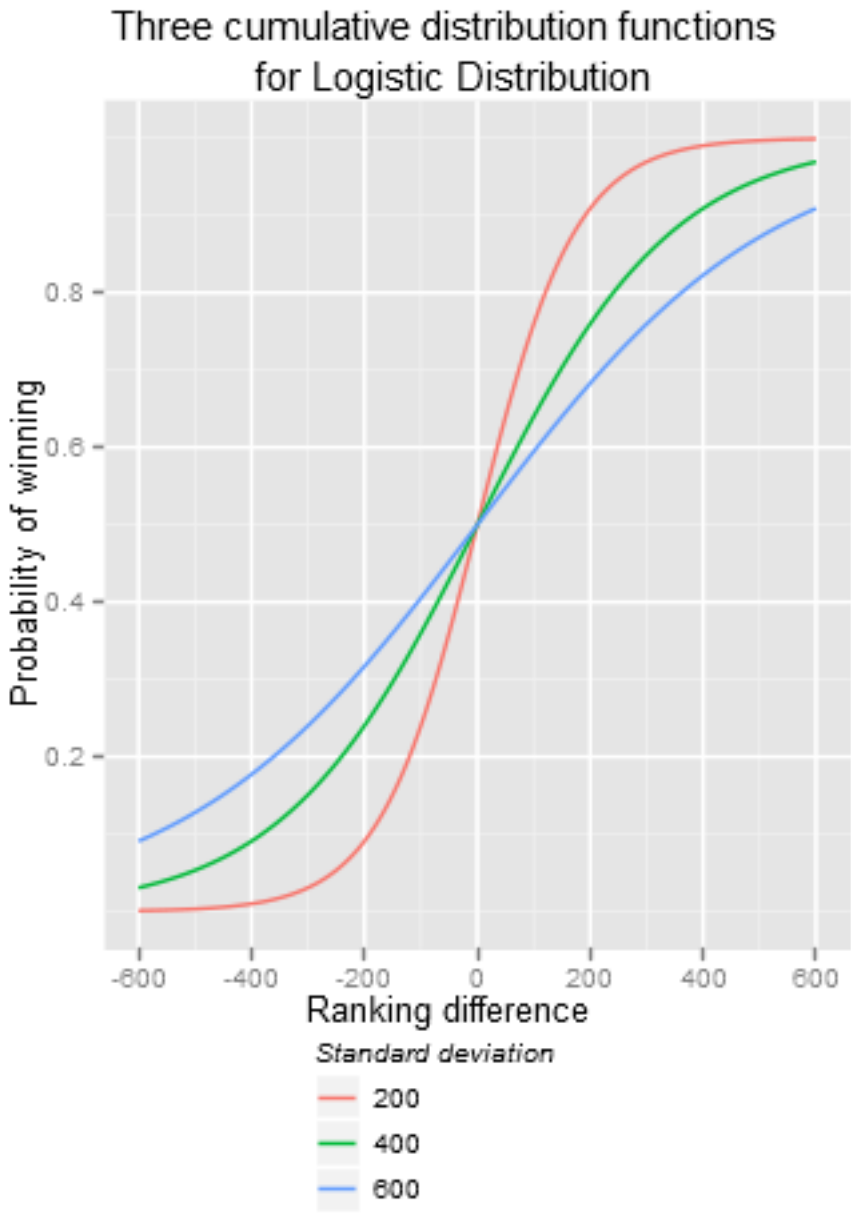

Figure 4.2: Graph shows how the logistic cumulative distribution function with mean zero looks for three different σ: 200, 400 and 600. The one used in the thesis is 400.

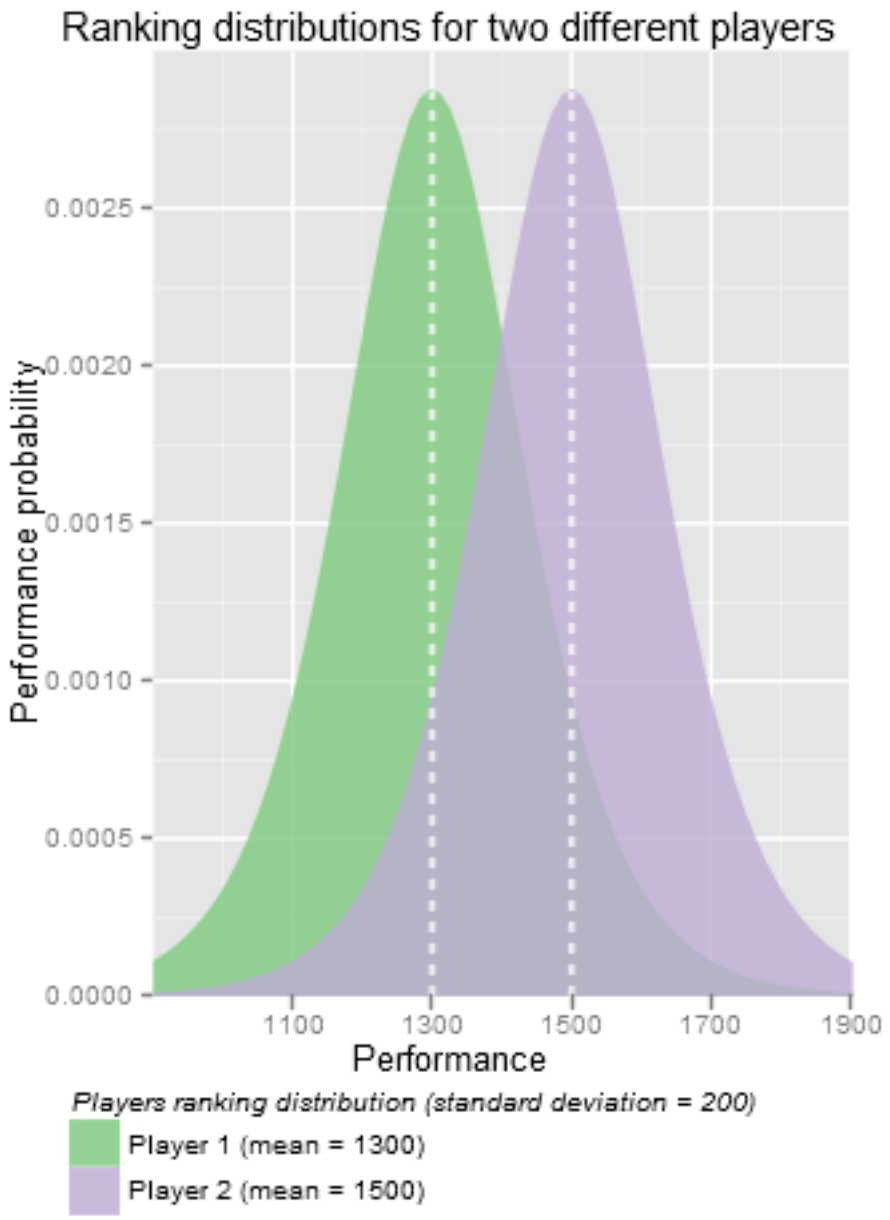

Figure 4.3: Graph shows the rating distributions of 2 players that play a match versus each other. One is rated at 1300 points, while the second one at 1500 points. Hence, the difference in their performance is equal to σ, which is 200.

than his opponent's. To get the better overview, player 1's distribution has been subtracted from player 2's, shown in Figure 4.4. The new mean for the obtained distribution is $\mu = 1500 - 1300 = 200$, and standard deviation becomes $\sigma = 2 * 200 = 400$. To calculate the probability that player 1 will win, one have to count the highlighted area to the left of $x = 0$. According to the definition of logistic distribution and putting $x = 0$, $\mu = 200$ and $\sigma = 400$ to the following equation:

$$P(p_1 > p_2) = \frac{1}{1 + 10^{(x-\mu)/\sigma}} \tag{4.6}$$

one would obtain equation exactly the same as Equation 4.4.
This shows the main idea of predicting the output with Elo and how logistic distribution is used to achieve this goal.

An important observation according to expected score is that when the rating difference between players grows to infinity (See Figure 4.2, the expected score converges to 1 (or 0). One should make a correction to this, because it is never true that there is almost a 100% chance for win/lose, even if world champions play versus beginners. The similar concept in chess is known as the "rule of 400", which says that the maximum rd used in calculations might be 400. If the difference between players' ratings is bigger than 400 (lower than -400) then it is artificially "rounded" to 400 (or -400). In practice it means that the maximum (minimum) expected score is around 0.91 (and 0.09). This rule has been applied to this model as well.

In chess the usual approach to enhance the rating system's predictions is to extend it with one additional parameter, which is responsible for slightly increasing chances of a player who plays with white. It is natural, because it was proved that the player who starts the game has an advantage. In bridge there are plenty of more factors, what makes it much harder to develop an accurate system. The following subsections introduce some of the most important factors that have to be taken into consideration, along with a proposition for including them into the model.

4.1.2 Rating of partner, opponents and players at other tables

The key question that occurs at this moment is how to satisfy the requirements listed in the previous chapter. First of all, the model is created for only two players and the number of bridge players in one deal is 4. Hence, the way to include partner's performance and second opponent has to be found. Moreover, the skill levels of all players at other tables should affect the predictions as well.

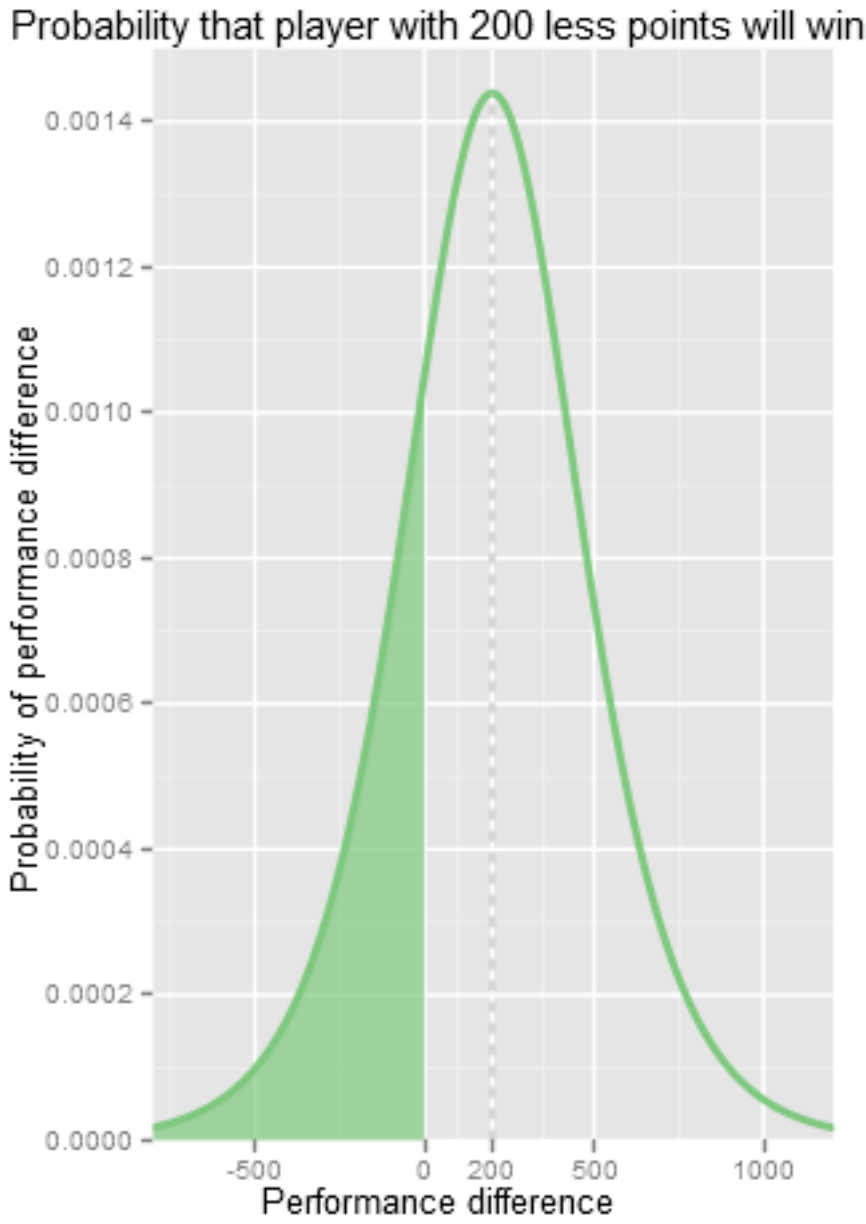

Figure 4.4: Graph shows the distribution of expected scores between two players whose rating difference is 1 class (200). The marked area is the probability that the lower ranked player will outperform his opponent.

For the first problem, two different approaches have been tested, in order to see which predicts better. One of the ideas was to generate artificial matches, based on one deal. The example should clarify this concept.
Assume, that in one game 6 tables are participating. Each of the tables consist of 4 players - two pairs playing versus each other. Instead of counting IMP averages, one can consider that each of the tables played 5 matches. It means, that Table 1 played vs Table 2, ..., Table 5 and Table 2 played vs Table 1, Table 2, ... , Table 5 etc. What is achieved, is that each of those matches could for a moment be considered as a separate events. One can then treat North-South from Table 1 and East-West from Table 1 as one team, that plays vs North South from Table 2 and East-West from Table 1 (An explanation for such definition has been given in Section 3.2. Because each player is equally important, one can just take the average of ratings from Team 1 and Team 2. The average ratings are applied to Equation 4.3 and this generates temporary points Δ for each player. The operation is repeated until no more matches can be created and all results are summed up for each player. The final point change for Player i the sum of all temporary points divided by the number of matches played.
The described approach satisfies all requirements related to players rating. The results were satisfying - at the top of the rating were people, who won a lot of matches and in total had a high plus imps score. On the other hand, the bottom of the table was occupied by players with very poor results. However, there was one theoretical bias, which lead to inventing another approach. The flaw is that too much weight is given for players at other tables. It is reasonable to claim that if there are only two tables in the game then it is crucial how corresponding teammates are playing. No matter how good score would be obtained at table 1, the second table could just generate very odd results, especially if one of the players is very low. However, the more tables are involved in the game, the less important is the skill level of players at other tables. It still matters, but in order to punish one pair, most of the others must generate odd results.
Such reasoning leads to another solution. Instead of creating artificial matches, a new parameter γ has been invented, which is meant to adjust predictions if one of the pairs sit in favorable lines. Such advantage occurs, when players at the opposite lines are stronger than ones playing at the same one. The γ is calculated for each table separately. For table t it is defined as follows:

$$\gamma_t = \frac{(R_o - r_{to}) - (R_p - r_{tp})}{2.5N} \tag{4.7}$$

where R_p is a total rating of players sitting at the same line (NS or EW) as pair for who the rating is being calculated and R_o is for their opponents. The r_{tp} is the average of ratings of pairs for who the rating is being calculated and r_{to} is the average of ratings of their opponents. The N is the number of tables, which

participated in the deal.
Interpretation of $\gamma > 0$ is that opponents have advantage, $\gamma < 0$ means that opponents are in a worse position and for $\gamma = 0$ no one is favorable to win. As intended, the weight is much higher for two tables than for 16. The γ should be applied to the main equation of calculating expected score as follows:

$$E = \frac{1}{1 + 10^{\frac{rd+\gamma}{400}}} \tag{4.8}$$

4.1.3 Difference in partners ratings

Another problem that should be addressed by the model and cannot be overlooked, is the difference in skill level in scope of pair. As for now, if pair 1 consists of players with ratings 1300 and 1700 plays versus a pair with both players with rating 1500, the win probability for both of them is 50%. The first impression was that the first pair is less probable to win. Such conclusion has been strengthen after imagining a more extreme example - the highest ranked player playing together with the lowest player against two averages players. Intuition was that the higher ranked player cannot use all of his knowledge - bridge is a partnership game and both players must be on the same page most of the time. Secondly, if the lower ranked player makes an error, there is usually nothing that the better player can do to fix it, especially in the bidding phase. A new parameter called ρ has been introduced, which was supposed to cover this situation. After a number of tests it turned out that data suggest a completely opposite solution - the first pair is more likely to win. Concept that might justifies that counter-intuitive behavior might be somehow similar with the idea about neighbors presented in Elo++ (Sismanis, 2010). A translation to bridge would be that if a player plays with a higher ranked partner it most probably means that his rating is *underrated*. It is reasonable if one assumes that good players play only with good ones. The assumption becomes weaker for intermediate players, meaning that it intermediate players are more likely to play with beginners. Hence, the skill level of a higher player should be taken into consideration while defining ρ. As for now, the equation is:

$$\rho = \begin{cases} 0 & \text{if } 0 \leq |rd_{\text{pair}}| \leq 20 \\ \max(\frac{|rd_{\text{pair}}|}{12} - \frac{\text{MAX}(r_{p_1}, r_{p_2}) - 1500}{10}) & \text{if } 20 < |rd_{\text{pair}}| \leq 400 \\ \max(\frac{400}{12} - \frac{\text{MAX}(r_{p_1}, r_{p_2}) - 1500}{10}) & \text{if } |rd_{\text{pair}}| > 400 \end{cases} \tag{4.9}$$

The notation has not been changed: rd_{pair} is the rating difference within pair, r_{p_1} and r_{p_2} are players' ratings, hence $rd_{\text{pair}} = r_{p_1} - r_{p_2}$. The difference of

20 proved to be a reasonable margin to tolerate and no penalty is given for it. However, the toleration increases along with lower rating, what can be noticed in Figure 4.5. The max() function has been put to avoid minus penalty. To sum up, the current form of ρ takes into consideration tolerance to low rating differences and the skill level of the higher player. However, more data should be gathered in the future to verify also the second case - when the rating should be lowered. Hence, the way of computing ρ is not optimal and should be further explored.

4.1.4 Partnership experience

The last adjustment that has to be covered by the model is the number of games played. For every game it is generally natural that the person with rating 1500, achieved after 800 games, is more probable to win than the opponent with the same estimated performance, but on the basis of very few games. However, one cannot forget that bridge is a partnership game. What really matters is not the pure number of games in general. Instead, one should focus on the number of games played with the current partner. However, not every game is equally important. There is a great difference between a pair who plays for the first time and the ones that have already played 100 deals, but there is almost no difference between partnerships with 1500 and 1600 games together. This suggests using some kind of logarithmic function to compute an additional parameter - λ. Number of tests were performed to find a value that best describes the obtained data, and the result was:

$$\lambda = \sum_{i=3}^{N_{p_1,p_2}} \frac{1}{\log_{10} 3i} \wedge \lambda \leq 150 \tag{4.10}$$

where N_{p_1,p_2} is the number of games played together. The first few games are ignored, because there must be at least little evidence that the two players indeed want to cooperate. Figure 4.6 shows how the logarithmic function used to calculate λ looks like.

4.1.5 Summary of rating prediction

To put everything together, the following final equation for predicting the score has been modeled:

$$E = \frac{1}{1 + 10^{\frac{((r_{o1}+r_{o2})/2+\lambda_o+\rho_o)-((r_{p1}+r_{p2})/2+\lambda_p+\rho_p)+\gamma}{400}}} \tag{4.11}$$

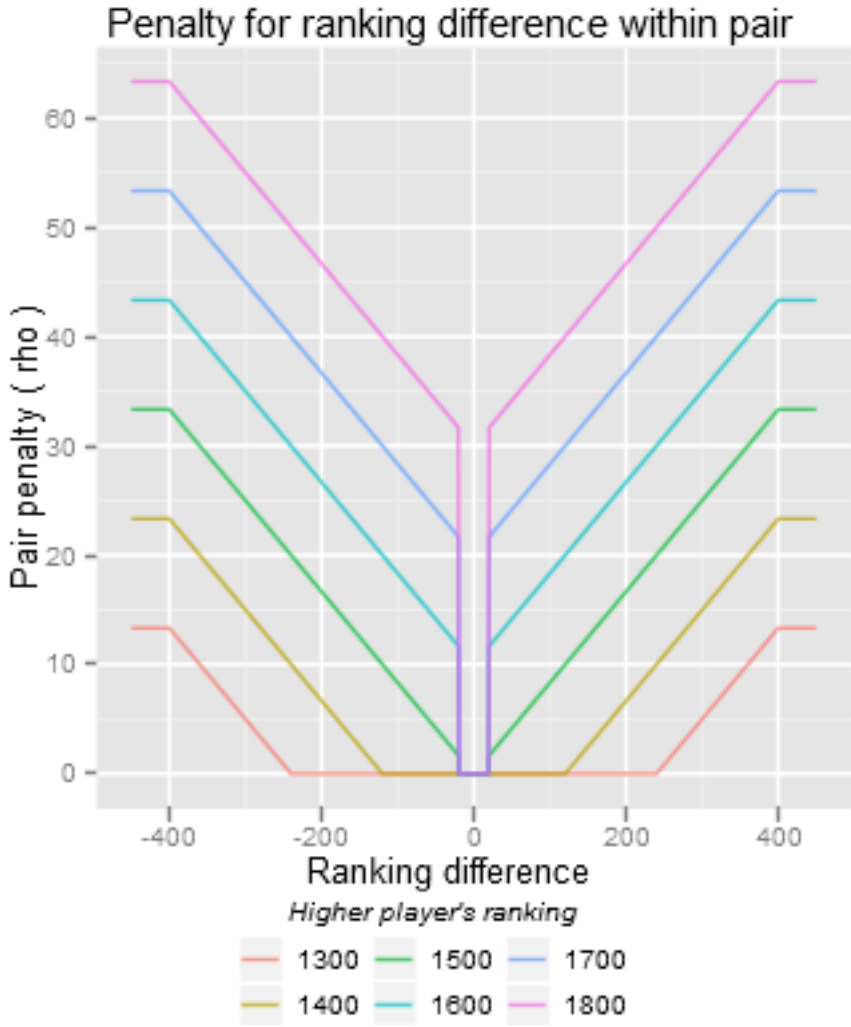

Figure 4.5: Graph shows ρ parameter, which is adjusting the pair average rating, depending on how big the difference between players is. The higher skill of the better player, the bigger adjustment is made. It is because good players tend to play with good players, which means that if their partner has a low rating, most probably it is underrated. In addition it takes into consideration what is the rating of a higher player, since there is a big difference if the higher ranked player is an expert or beginner - for the second one the reasoning is much weaker.

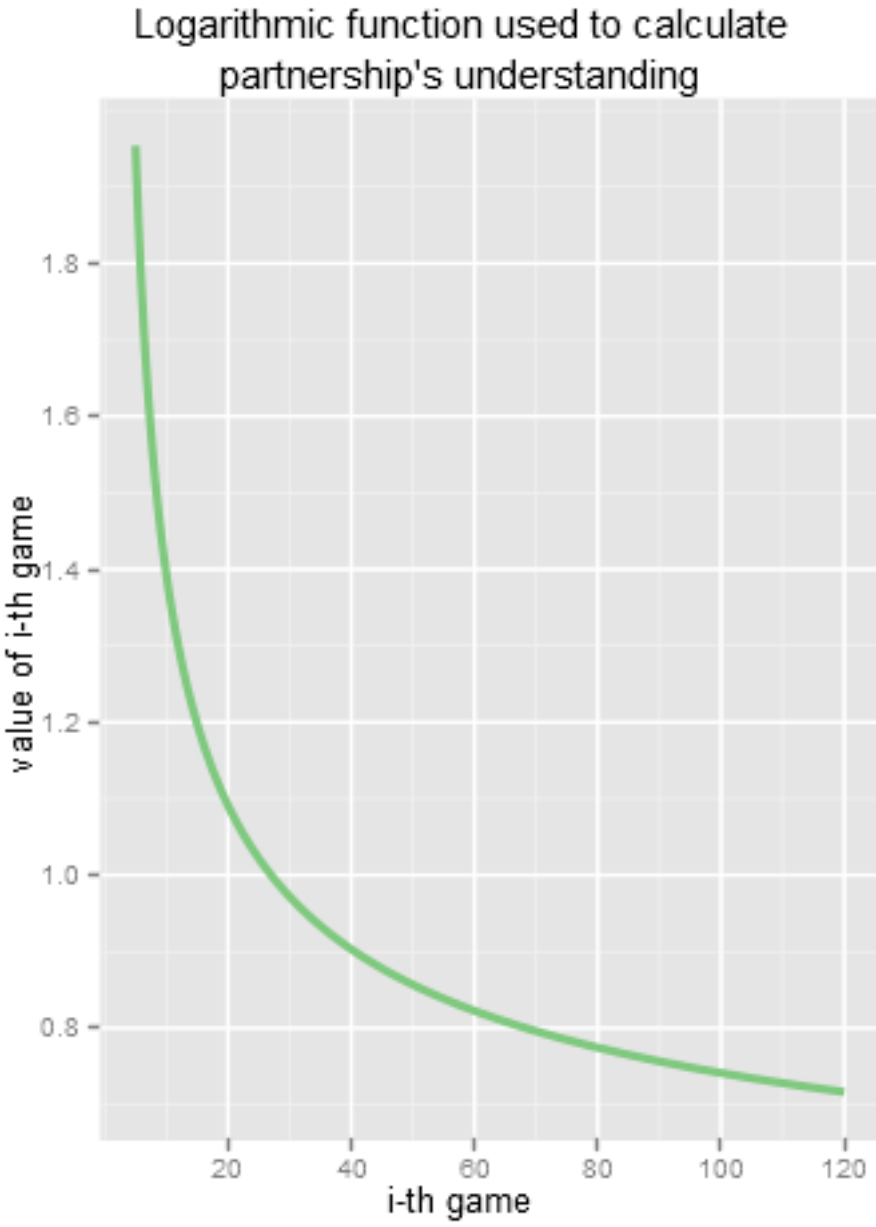

Figure 4.6: Graph shows how the value of i-th game counts less to λ - Partnership understanding coefficient.

The parameters that are taken into consideration during calculating the predicted score:

- Skill level of the partner and both opponents - the ratings of each pair are averaged
- Skill level of all other players via γ parameter
- Difference between partners skills - ρ
- Number of games that partnership has played together - λ

4.2 K factor

Having a formal way of calculating expected scores, one can proceed to defining the second and last part of the model: The K factor. There might be various interpretations of it, the most important being:

- Weight of how important the game was
- How big an influence should the game have in modifying players rating
- How much trust do we have in the player's previous rating
- What is the maximal/minimal amount of points that can be added/deducted
- How fast player is believed to make progress

However all of them lead to one, very specific goal: optimize future predictions. It is important not to set it too high, otherwise each player's rating will be changing all the time and will never be established. On the other hand, if it will be too low, then it will take too much time to arrive at the proper value.

4.2.1 Defining K

The process of choosing the right default value of K starts at drawing 6 functions (Figure 4.7) for various values of K to see how the points change Δ is affected by the rating difference rd. An immediate conclusion is, that when the rating difference is 0, meaning both players have the same probability to win, the

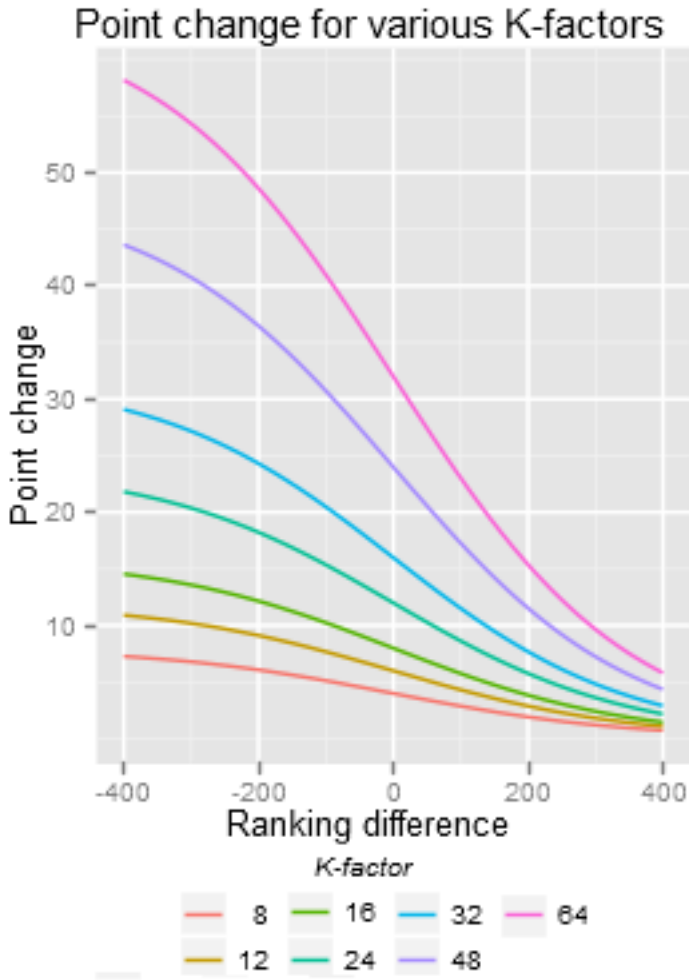

Figure 4.7: Graph shows how the rating difference affects the amount of points that are gained by the winner for different K values. On x axis there is rating difference rd and the function value can be derived by putting the argument to Equation 4.3 and its result together with $s = 1$ into Equation 4.1. The reason of choosing an argument range from -400 to 400 is that these are the only values for rating difference, as explain in Section 4.1.

number of points to be gained (lost) is $K/2$. On the other hand, if one player had 100% chance of winning, then the Δ would be exactly K[1].

Further research showed that the K-factor (Ralf Herbrich and Graepel, 2007; Moser, 2011) can also be written as:

$$K = \alpha\beta\sqrt{\pi} \tag{4.12}$$

Where α is the *trust* to the new rating and β is standard deviation, which can be interpreted as a skill width (Moser, 2011). Since the β has been set to 200 (as explained in Section 4.1), one can generate a plot of what trust corresponds to various K: In the case of chess, for default Elo, the reasonable and often used value is $K = 24$ - which is assigning about 7% trust to the new rating. Based on experience, one could definitely say that this value is too much for bridge. The reason is that in chess the only additional parameter for predicting score, beside opponents rating, is who played with white. In bridge on the other hand, it matters whether a player played with his regular partner (if he has one), what was the skill of the partner, what is his skill level, how much difference there is between him and his partner, what is the difference of the skill level with opponents, are the opponents regular partners or not and so on. One can end up with a really long list of other factors, which makes the case very complicated. Trying to define optimal an K-factor manually would not be possible to do. Hence, many experiments were run in order to find a value that works best for sample of $250,000$ deals. The K-factor that has given the lowest prediction error was very low - about 2. It is not probable that such low value should really be assigned, since the point change is very low in such case. Most probably the result is related in the known problem with optimizing rating systems called overfitting (Sismanis, 2010). Finally, the value of 4 was chosen, since it provided a good trade-off between dynamic and accuracy.

4.2.2 K-factor modified - K_d

Another usage of the K-factor is to satisfy requirements about not only who won, but also by how much. One can do it by increasing K if the score difference was high and make it lower if the match was close. Such adjustment has been already used by World Football Rating (Runyan, 1997) - they modified K on the grounds of a goal difference in a match. In bridge, judgment whether the match was one-sided or not depends on the scoring type. To remind, the one used in this thesis is IMPs (See Section 3.2). It means that it varies from 0 to 24. Figure 3.1 might suggest that only integers are possible, however one

[1] It will never be the case that winning probability will be greater than about 91% due to applying the rule of 400 - See 4.1

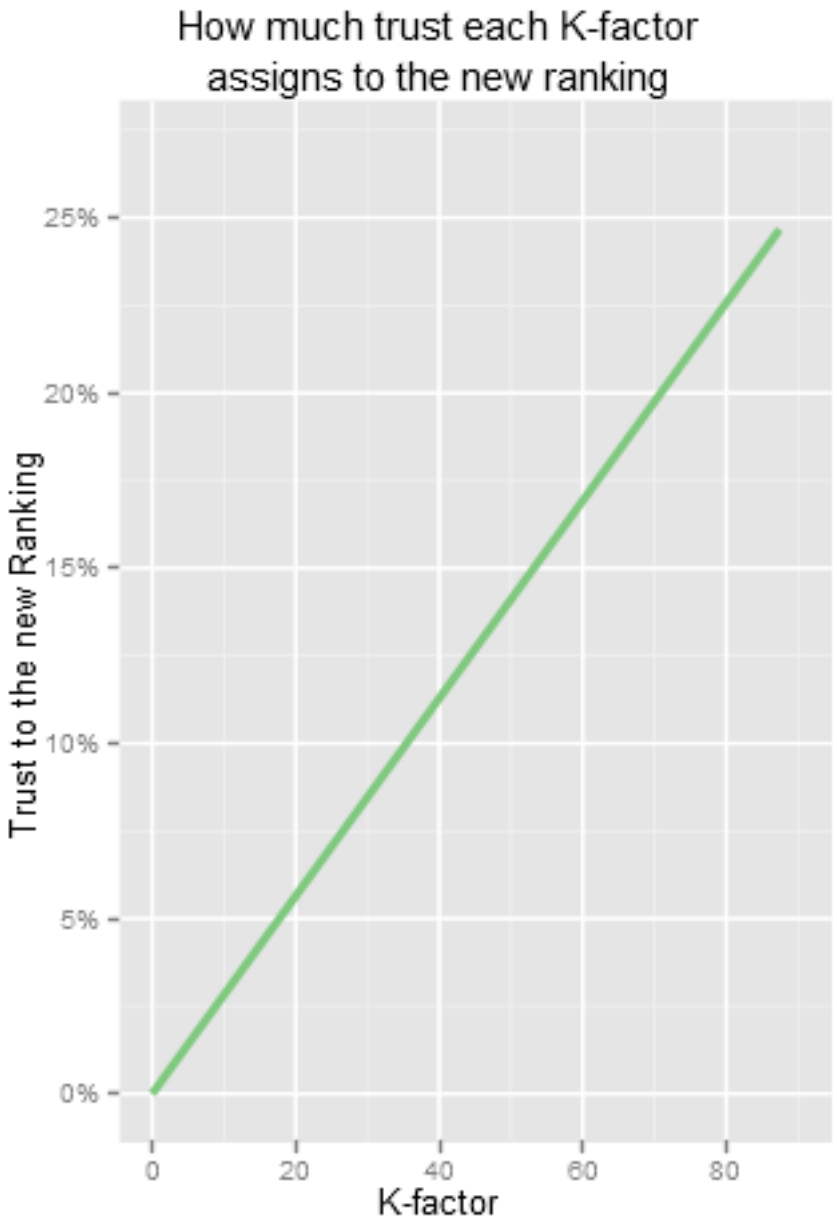

Figure 4.8: Graph showing how much trust each K-factor assigns to the new rating.

should remember that an average is taken. The decision was made to group all possible outcomes into five categories. The first one is when the score difference is very small - less or equal one IMP. It is considered a draw, as explained at the beginning of this Chapter and hence the K-factor will not be modified for this range. The next group contains scores less than 5 IMPs. It is considered a very close victory. The next range is from 5 to 13 imps and it defines typical situations for treating it as a significant win over opponents. If the score was between 13 and 19, it means that opponents have been dominated and hence is a little bit more important than the previous. The last range - 19 and above - shows complete out-performance. The reason for such ranges lies in the bridge scoring. There are some typical situations in which the final score falls exactly to the given sets. Unfortunately there does not exist (and cannot be provided) any mathematical explanation of them. It is a subjective element that I introduce to the model, however I base it on over five years experience of playing bridge. Based on the explanation above, the mathematical definition of m, which can be described as K factor modificator, for each possible value of score difference D (measured in IMPs) is defined as follows:

$$m = \begin{cases} 1 & \text{if } 0 \leq |D| \leq 1 \\ (0.5 + (|D| - 1)/50) & \text{if } 1 < |D| < 5 \\ (1 + (|D| - 5)/50) & \text{if } 5 \leq |D| < 13 \quad \text{where } 0 \leq |D| \leq 24 \\ (1.6 + (|D| - 13)/50) & \text{if } 13 \leq |D| < 19 \\ (1.8 + (|D| - 19/50) & \text{if } |D| \geq 19 \end{cases} \tag{4.13}$$

The base and denominators have been chosen through the number of experiments and they were proved to reduce the error in predictions.
The visualization of how m increases is shown in Figure 4.9 (the draw scenario has been excluded from the chart). Such modifications are done for each game separately. Since in the rating, each of the players will play many games, it is necessary to keep track of each games score. The simple approach of taking an average of all K-factor calculated for each game is however erroneous, and leads to very wrong behavior of the rating system. If a player has lost 10 games by 20 imps, but he won the next 11 games with only 2 imps, he will gain a high boost even though he should not. As a solution, the rating system keeps track of two different sums: for winning games and loosing games. Additionally, to distinguish if the good / bad score was gained versus a good or bad player, a weight is assigned based on expected score versus opponents. Finally, if many games were won (lost) high, and all the others have been lost (won), but were very close, then the penalty should be lower than if the won (lost) matches were also very close. The final way of computing the K-factor for the rating period is:

$$K_{\text{positive}} = \sum_{i=0}^{N} W_i * K * d_i - \sum_{i=0}^{N} (1 - W_i) K * (\frac{1 - E_i}{2}) \tag{4.14}$$

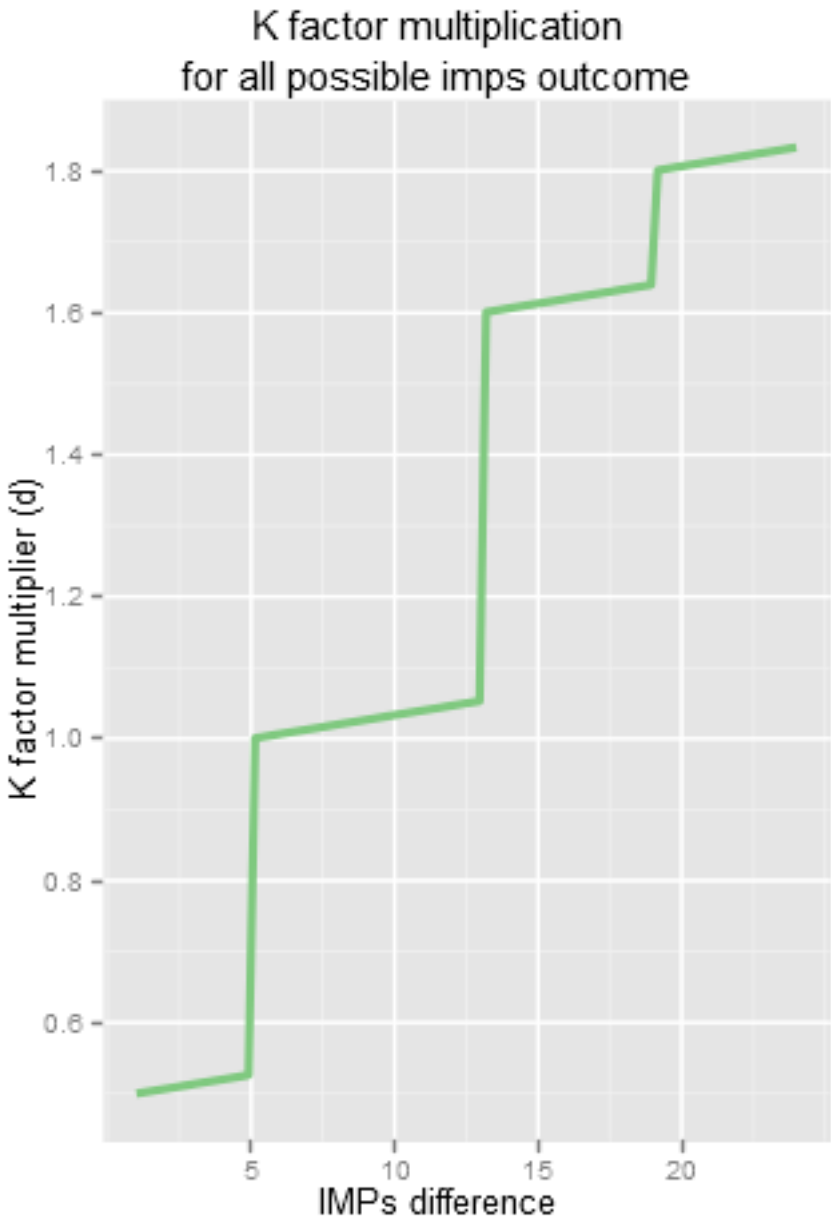

Figure 4.9: The graph represents how the K-factor is modified based on score difference, which is measured in IMPs. The x axis represents whether the match was close or not. The higher argument, the less close and the more dominant were the winners. The y shows the value of the K-factor multiplicator - d.

$$K_{\text{negative}} = \sum_{i=0}^{N} L_i * K * d_i - \sum_{i=0}^{N} (1 - L_i) K * (\frac{1 - E_i}{2}) \tag{4.15}$$

$$K_d = G * \frac{K_{\text{positive}}}{\sum_{i=0}^{W} W_i} + (1 - G) * \frac{K_{\text{negative}}}{\sum_{i=0}^{L} L_i} \tag{4.16}$$

The W_i and L_i are either 1 or 0 and they symbolize win and lose. The following is always true: $W_i \oplus L_i = 1$ (draws are ignored). The N is the number of games played by the player, E_i is expected score and d_i is the multiplicator obtained from Equation 4.13. The interpretation of the first two equations is that whenever a player wins a game ($W_i = 1$ and $L_i = 0$), his K_{positive} increases, while the K_{negative}. If the opposite is true - the player lost a game - then analogically the negative sum is increased, while positive decreased. The value by which the second sum is lowered depends on the expected score and the imps - the more likely the player was to win, the less modification is done and the more imps are gained, the higher change in rating is possible. The final K depends on G, which is either 1 or 0. The first option means that the player overperformed at the end of rating period - his real score S was greater than the expected score E. In that case, the positive sum of K is taken into consideration and negative does not count anymore. If the player underperformed, then the negative one is taken.

4.2.3 Provisional and established players

Clearly it would not be right to set one global K for each player. For the special treatment especially deserves two types of players. The first one is the very new player, who did not play many games. Generally their start-up rating should be trusted much less than the same rating for players who played several hundreds games. That is why it is generally encouraged to increase K for so-called "provisional players". What seems to be the most reasonable choice, is to make the K factor really big for the first game, and then step by step decrease it with every game until users have play enough games to no longer be treated as provisional players. For the purpose of the thesis 10 has been chosen as a sufficient value. The way of modifying K is defined as follows:

$$K_p = 4 + \text{round}(\frac{10}{N}) \tag{4.17}$$

For the first game, the K value for provisional players - K_p - will be 14, the second one 9 etc.

On the other hand, it is very common that experts, who achieved a relatively high skill, tend to improve much slower than the others. That is why it is reasonable to put more trust in their ratings and prevent too large rating changing by lowering the K-factor if they reach certain rating. As alternative, some chess federations decided that the K should strictly depend on the number of games that a player has played - the more games, the lower K. Such concept seems also reasonable, however it is not clear how to apply it to bridge. More important than total number of games is how many games have been played with a regular partner. However many players, even those who have one partner with whom they play the most important events, have many others partners as well. It would be unfair to punish players for playing with weaker and new partners, on the other hand player's skill level should not only depend on the real partner. Moreover, there is no clear way of how to find player's regular partner - the intuitive metric which is the ratio of games played with each player is not very reliable. Hence, the first approach has been used, and the K-factor is modified based on the player's rating. If the player ever reached a rating of 1700, his default K value becomes $4/1.5$, which is $2.(6)$, and if he reached 1900 then it becomes $4/2 = 2$.

4.2.4 Final K

To sum up, the K-factor's default value is 4, which corresponds to about 1.1% trust in the new rating. If the player have not played 10 games, then his K is artificially increased. On the other hand, if at least once in his career, a player reaches 1700 or 1900 rating points, then his K factor is lowered, respectively, by a factor of 1.5 or 2. Additionally, the K-factor might be increased or decreased, depending on how close a player's matches were. The system rewards high wins and punishes high loses by transforming K into K_d.
As it can be noticed, the K-factor does not depend on the number of games, but only on ratings. It is because of the complexity produced by involving partnerships instead of individuals. Any trials of involving number of games into a definition of the K-factor drastically increased prediction errors instead of lowering them. Even though it is probably possible to do, the focus has been put on adjusting parameters for rating predictions.

4.3 Summary

The final model is an extended version of the Elo model and adjusted to the nature of the bridge game. The rating period has been set to three days, meaning

that the expected scores and real scores are stored temporarily and used in the final calculations. New players' rating can be mathematically described as follows:

$$R_n = R_{n-1} + K_d * (S - E) \tag{4.18}$$

Interpretation of S did not change at all, and it is still 1 for win, 0.5 for draw and 0 for lose. The only modification that had to be done is treating any result not greater than 1 as a draw. Many more changes were introduced to the way of calculating the expected score E. First of all, "player's rating" from the Elo point of view (which is the base to predict a score) is an average of the ratings of two partners. Moreover, there are three additional factors that should help predict the output of the game. First of them is γ, which defines how big an advantage or disadvantage the opponents have, due to the strong partners and weak opponents. The next parameter λ is calculated separately for both pairs and it estimates how well partners understand each other, basing on their game history. The last modification is done by ρ, is also calculated for each pair separately. It is assuming that if one of the partners has weaker score, it means that he is underrated and the probability of winning should be slightly increased. The final equation for expected score E has been introduced in Equation 4.11. The K-factor is modified based on how high wins / loses were and versus who they were encountered.

Chapter 5

Applying the Model

Once the calculations were done, the results have been stored partially in the database and partially exported to the CSV[1] to make it easier to visualize various features of the data. Most of the plots described in this chapter have been performed using open source and a very powerful statistical platform - R - along with many external libraries, from which the most important one was ggplot2 (Wickham, 2006).
The chapter begins with a short section about the data set, which contains information about the first three steps of the visualization process (Fry, 2008): acquiring, parsing and filtering. Next, the measurement of error prediction is visualized and commented. The last section contains a visualization of players' statistics.

5.1 The Dataset

The data were supposed to be delivered by a company owning software, which is used by the great majority of all bridge players, including world champions. There are a few thousand of simultaneous players On-line, who produce plenty of real life data to analyze. Unfortunately, due to the communications issues it

[1] CSV stands for Comma Separated Value

was necessary to implement web crawling technique (Segaran, 2007) in order to obtain necessary statistics.
The chosen approach was to recursively web-crawl a free accessible public statistics archive located at http://www.bridgebase.com/myhands/index.php. The method can be summarized with a very short pseudo code:

```
startupPlayer = "player"
while playerName do
        getStatisticsForPlayer(playerName)
        parseAndStoreStatistics(content)
        storePlayersNames(content)
        markPlayerAsChecked(player)
        playerName = getFirstUncheckedPlayer()
end
```

However, this algorithm in pessimistic scenario does not ensure obtaining the gaming history of all players. One consider a graph, where a node represents a player and an edge between two players indicates that they have played together at least one game. Then, if there is no path between two players, one of them would not be found. However, the results indicate that this is not the case for larger communities, because for the time span of 82 days there were over $25,000,000$ million games records obtained for more than $200,000$ players. It might be the case that a very small community has not been detected. However, extending the algorithm by checking an additional condition, which would ensure obtaining data for all players, would result in increasing the number of HTTP requests by the number of deals - $2,000,000$. The simple approach needed two weeks of data gathering for a timespan of 30 days and it was necessary to perform only as many queries as there are unique players - about $200,000$.

The parsed data that have been used to apply the model described in the previous chapter are:

- Time at which a deal has been played - it was used to determine to which rating period the game belongs.
- Names of 4 players that played deal
- Points achieved from a deal
- Id of the hand - to be able to know which hands should be grouped together and compared to each other
- Analysis name - it stores information if the points are given for EW or NS. If the name of the player for whom statistics were crawled was not

stored, all the results would be useless, because it would not be possible to define a winner.

The process of acquiring the data was unexpected and time-consuming, however the relatively simple approach turned out to be very efficient. After two months of acquiring data, this process was stopped and the last step has started: filtering the data. Unfortunately, this process required deleting a significant amount of the dataset, however without it, future output would be erroneous. The first thing that needed correcting was a hidden issue, which was not spotted during the process of web crawling. It probably occurred because of using a personal notebook instead of a dedicated machine. Due to the very long time required to accomplish the task, the necessity of using the computer for other purposes interrupted the process as the application was forced to pause, and some players that participated in the game have not been stored in the database. This had the unfortunate effect of having to remove about $2,000,000$ games, which were all missing one player. The decision to remove them was not made immediately, however after many trials and failing experiments because of too much noise, it was decided to do so. Moreover, it was also necessary to delete the hand records for games, which contained a GIB[2]. The system from which the data has been obtained allows robots to play at three positions simultaneously, however it does not provide any way to distinguish between them. As a result, the interpretation is that GIB might play versus himself. Beside those two cases, during the data analysis it turned out that the platform from which data was obtained has a quite serious implementation error, which rarely allows players to play the same hand two times. Such situation should not be possible and erroneous deals have been deleted. Last but not least, all deals, which were played at only one table, have been removed from the database. The reason why such deals do not provide any meaningful information and hence can be deleted is that there is no other players to compare the result (See Section 3.2). This issue concerned mostly the deals that have been downloaded at the end of the process. The final amount of games that were used in the model is $21,684,154$ ($2,226,279$ deals) played by $203,279$ players.

5.2 Accuracy and RMSE

The metric that was used during the accuracy measurement is called Binomial Deviance. It is given with the following equation (Sonas, 2011):

$$D = -[S * \log_{10} E + (1 - S) * \log_{10}(1 - E)] \tag{5.1}$$

[2]GIB is the name of the robot that might be hired by the players who want to practice certain elements of a game.

It was computed for each game for N-S line. The S is the real score and E is the expected score. Once the Binomial Deviance is calculated, the average is taken. The lower the value, the more precise the system is. In order to obtain accuracy in percentages, one should just put 10 to the power of $-D$ (Sonas, 2011):

$$A = 10^{-D} \tag{5.2}$$

Besides Binomial Deviance, a second metric - RMSE - has been computed simultaneously. Its definition has already been introduced in equation 4.2. To verify the correctness of D, a simple test has been performed. A null system has been implemented, meaning that for each game the expected score was always 0.5. It was run on the a sample consisting of 230000 deals. The obtained D was 0.30102999566567600, what after applying to Equation 5.2 gave result 0.49999999999, which can be easily rounded to 50%.

The total accuracy of the system after processing all deals in the system is 50.018%. It is not a shocking improvement, however if one takes into consideration all factors that occur in bridge, any improvement of the null system might be considered a success. However, it would be a mistake to claim that the system works perfectly only on the basis of this metric. A few additional visualizations have been performed to verify its correctness.
First of all, the accuracy and RMSE for each period have been plotted. Theoretically, the perfect rating system would always enhance its predictions, meaning that there should be a negative correlation between the number of periods and predictions error. Both measures were started after period 2 to reduce noise in the data. Because all of the players in period 1 have the same rating, all of them will have the probability to win 50%, hence it will not provide any useful insight into the data - only noise. The second period is ignored as well, to let the system make corrections if necessary. The results are shown in Figure 5.1 and Figure 5.2.

The results are not perfect, however they are not very bad either. One can see that the system tries to act as desired, however it has some serious problems with few last periods. There might be many explanations for that. It might be because of the quality of the data gained for the last time span, which might contain much more noise than for the previous ones. It could easily result in poor predictions. The other reason might be that players are unstable - in some periods they win a lot and then they start to loose, what confuses the system about theirs true skill.
After seeing how the accuracy and RMSE look for each period, it seemed interesting to verify how the system predicted the games results for each player. To filter some noise, the requirement has been made, that a player had to play at least 10 games in a period to be taken into consideration. The results are presented in Figure 5.3 and Figure 5.4

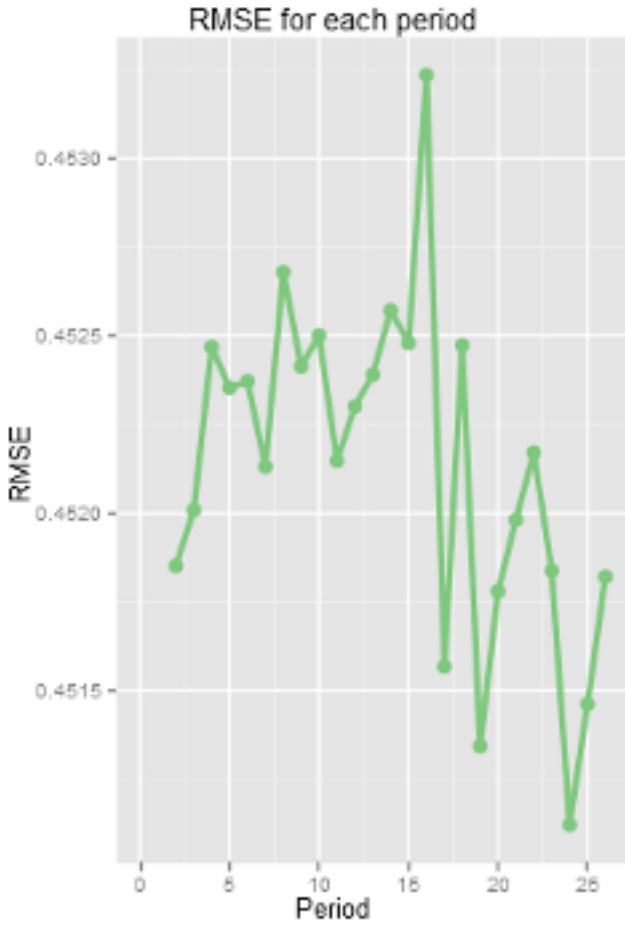

Figure 5.1: Figure shows RMSE for each period starting from 2. The desired situation is would be if there were a very strong trend of lowering RMSE. Unfortunately, it hardly can be said about this figure.

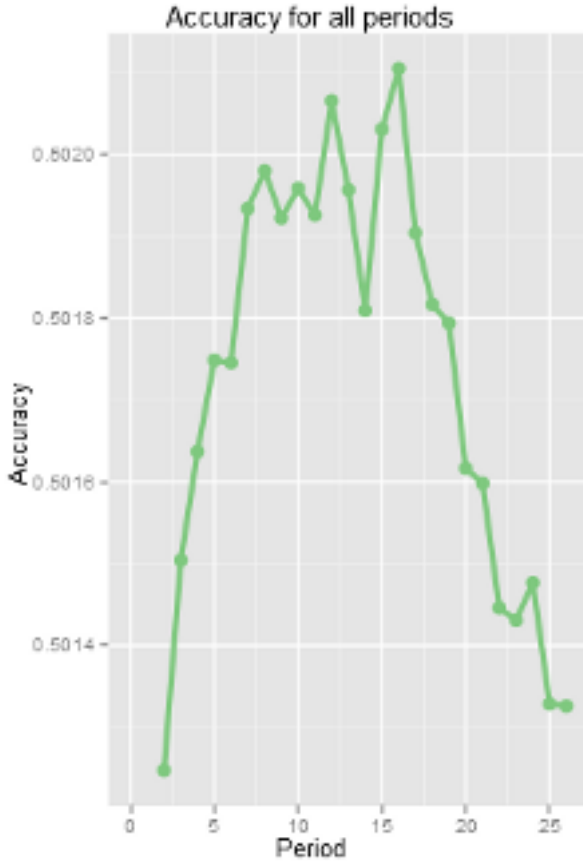

Figure 5.2: For each rating period starting from 2, the accuracy of the system has been measured. The system acted as desired to some point, however the last few periods had a much lower accuracy. However, it should be noted, that each of the period had total predictions greater than a null system would provide.

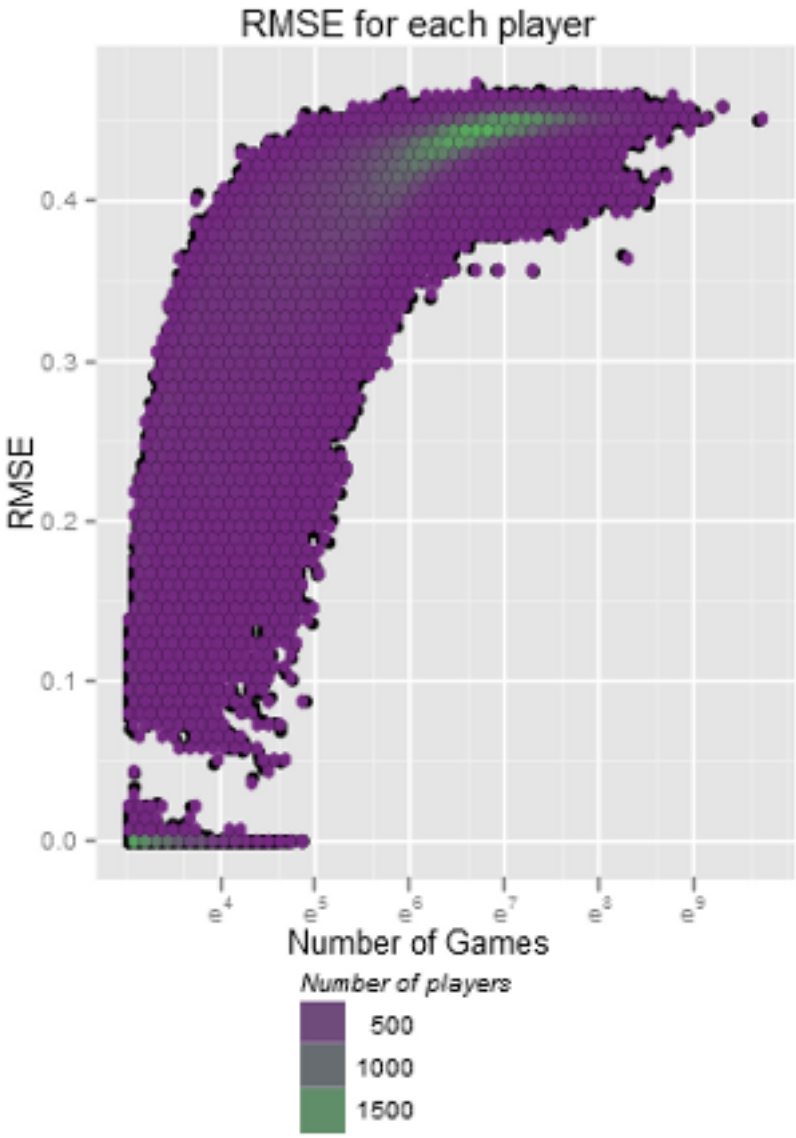

Figure 5.3: The plot shows how well the system predicted the output of players games. In addition, it shows how many games the player played. The color indicates how many distinct players who played the same amount of games had the same RMSE. One can see a very heavy trend in area between e^6 (403) and e^8 (2980) games played. The more games were played, the worse prediction - RMSE was bigger.

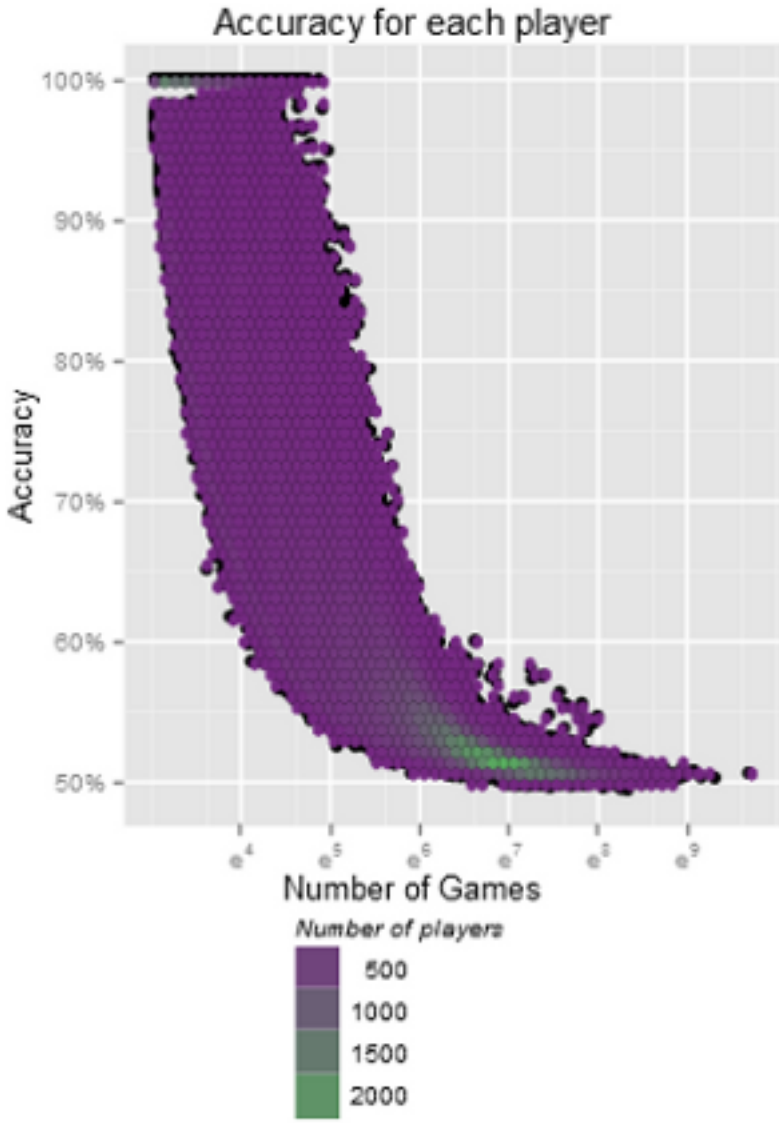

Figure 5.4: The figure visualizes what was the accuracy for each player. The x axis is the number of games played by the player. The additional color is there to show how often predictions happened for a given number of games. For a low number of games there were incredibly high accuracy for a lot of players - 100%. However, the more games players played, the more accuracy dropped. Most of the players played from e^6 (403) and e^8 (2980) games and there is a very clear trend for them as well.

The trend is extremely clear in both figures - the more games are played, the poorer the results. There are two characteristic areas. The first one is for players who played only about e^3 (20) games. There is quite a lot of them and for almost each of them the accuracy was about 100% and RMSE close to 0. The second one is area between e^6 (403) and e^8 (2980). One sees that it is very typical for all players near e^8 to have a lower game predictions accuracy than e^6. The fact that outliers do not exist, is very interesting. Only a few of them occur between $e^{6.5}$ and e^8 in the accuracy plot and some between e^4 and e^5 for RMSE. This might indicate that there is a systemic bias in the model. The good thing about these plots is that it does not seem that if players would had played more games than they actually did, it would result in significant lowering accuracy. This assumption is reasonable, especially if it will be taken into consideration that it is already a logarithmic scale and the players that played most amount of games break the trend.
The reason of obtaining such trend at the beginning is very clear - the more games, the harder it is to predict all of them, which would result in error. However, at some point this pattern is expected to be broken and the accuracy should start rising a little. One reason why it does not look like that is that bridge is a game of chance and even good players have to lose a lot, because there is no other possibility. This results in high expectations for good players, which does not come true, what lowers the accuracy. In the next period, after lowering the expectations, they actually had very good session what makes the process repeatable for many players.

Another plot that is worth to see is a boxplot for accuracy for each player for each period. Its interpretation is that at least 50% of all observations lie inside the box - between Q1 and Q3. The median (Q2) is represented by a horizontal line within the box. It has been shown in Figure 5.5. An important conclusion from this plot is that only maximum 25% of players have an accuracy lower than 50% - however usually it does not drop below 40%. Accuracy for all other players varied from 100% to about 50%, which is likely to be due to the high variance of numbers of games played within the period. As shown in the previous two figures, players with a low amount of games usually have a 100% accuracy, and the more games, the more it tends towards 50%. It is also important to see, that periods (starting from 2) do not really differ from each other. There are some small changes, but the general behavior is all the same.

For the completeness of the analysis, the correlation coefficients between accuracy, RMS, Period and Games have been calculated. The visualization of them is shown in Figure 5.6. Two choices of a way of calculating correlation were considered - Pearson's and Spearman's coefficients. The second one has been used to check where correlations exist and where not (Bolboaca and Jantschi, 2006). The minimum value for correlation is -1, which means that there is perfect inverse correlation - higher arguments produce lower y. It is represented by

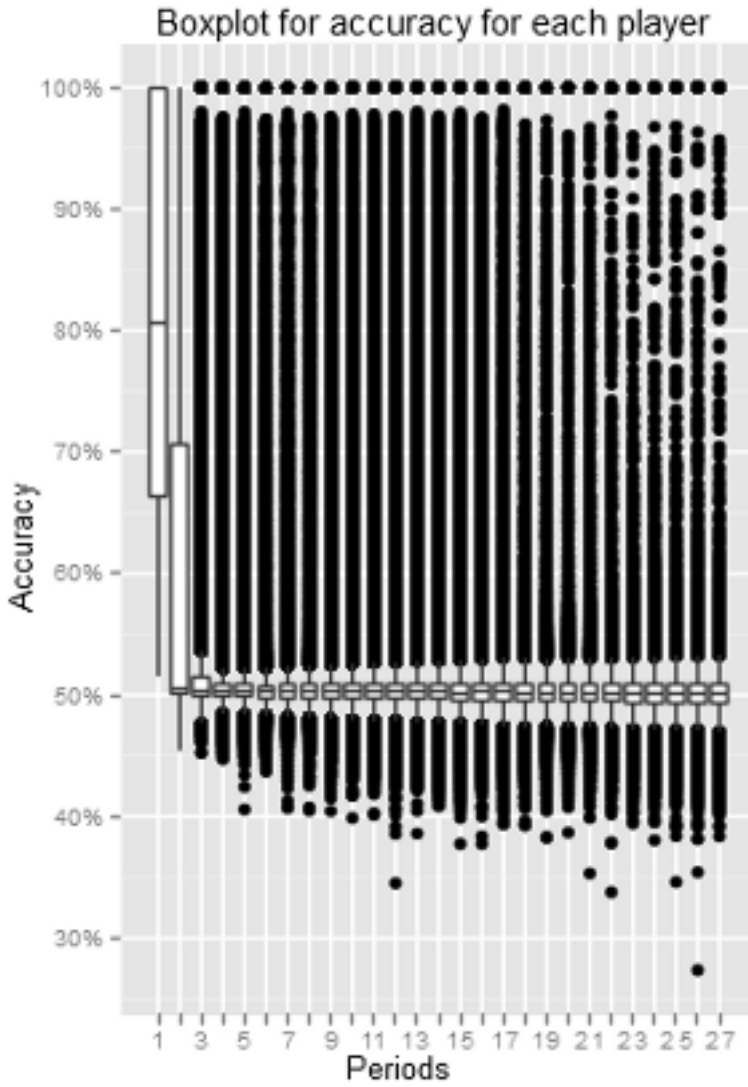

Figure 5.5: The created boxplot shows accuracy computed for each single player, grouped by period. It shows that for each period there are a lot of of outliers. An interesting fact is that nearly all of the vertical line is covered by outliers. The reason to this is very likely to be the high variance of the number of games played within period and the correlation between games and accuracy seen in the previous plots.

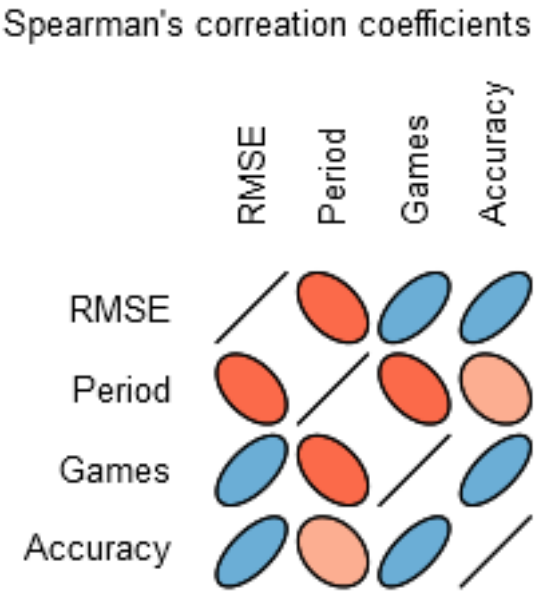

Figure 5.6: Visualization of Spearman's correlation coefficients. The blue ellipses indicate positive value, while the red are negative. The more the ellipse looks like a circle, the closer value it is to 0. It confirms that there certainly is positive correlation between Games (number of games played) and RMSE accuracy. There is also small, but still, negative correlation between period and accuracy.

red, skewed in the left side line. The highest is 1, which means that pattern of getting higher y for higher x is perfect. It is represented by the blue line, skewed in the right side. The value 0 means that no pattern could be found and it is represented as a white circle. The result is shown in Figure 5.10 As expected, the positive correlation between RMSE and number of games exists, as well as a negative one between accuracy and periods.

To sum up, the results are not really great. There is a lot of noise and chaos during predictions. The fact that number of games and periods are strongly (negatively) correlated with accuracy is not a good feature, since the opposite was desired. There is however very good justification, which should be stressed again. In bridge, plenty of different factors matter. Including and optimizing all of them is certainly far beyond the scope of the thesis. What would probably have resulted in an increase in relatively high accuracy is using more advanced optimization methods, like for example the stochastic gradient descent tech-

nique (Spall, 2003), which has been successfully been adopted for the current best chess rating system called Elo++ (Sismanis, 2010; Sonas, 2011). Counter-intuitive was the fact that reducing duration of the rating period - what results in analyzing less amount of games during each period - did not provide any boost to the prediction and all diagrams showed in the section were very similar. There is one more reason of why there might be a negative correlation between period and accuracy - it is the fact that different players might play in these. What should have been done is choosing a sample of all players that have played in the same periods and then compare the results. The assumption that the noise would average out most likely does not hold for this particular data set.

5.3 Players' Statistics Visualization

The following statistics about player have been measured about each player:

- Name - Players name
- Ranking - Players rating
- Periods - How many distinct periods has a player played
- Games - How many games has a person played
- RMSE - What is the total average error of predicting the score calculated by dividing the difference between Total Real and Total Expected by Games
- BD - another metric to calculate average error of predicting the score
- Wins - How many times the player won
- Loses - How many times the player lost
- Draws - How many times there was a draw
- Win ratio - How many % of the games have been won
- Imps - What is the total number of imps gained (lost)
- Partners - With how many different partners a player played real score

The aim of the first visualization is to present an overview of the statistics of top and bottom players. One way of doing it is to draw a table and put all the numbers there. However, even though such data presentation will be useful, it will not be very usable to find data patterns and grasp a fast overview - especially, if table will contain many players. As it is not important what the exact values are, but more so, their distribution, it was decided to create a heatmap. The result of the visualization can be observed in Figure 5.7. There is one very clear pattern - players with a high rating also have a lot of imps, while lower ranked players have less. This is a very desired behavior, because what distinguishes a good player from a bad one is the total number of imps in the long run. This indicates that the right players are at the top and bottom of the leaderboard. Another observation is that there might be a case of the players being penalized too heavy. It is because there is a much clearer distinction between top and bottom players on the grounds of number of loses than based on wins. Finally - which is a relief - the rating does not seem to depend on the number of games nor periods. The previous visualizations showed a very strong correlation between accuracy and this value, which could have very strong consequences, directly polluting the final results.

The next visualization will show how ratings are distributed between players. In related works (Sismanis, 2010; Glickman, 1995) the histogram was used to show the rating distribution and hence such approach has been chosen as well. However, not the whole subset of the data will be taken. There is a lot of noise generated by players who played very few games, which is why only those players who played more than 70 games in total will be taken into account. This requirement goes for all other visualizations as well. The obtained histogram is shown in Figure 5.8.

As expected, the most common rating is the default value - 1500 - and ratings near it. What is interesting, is that it is very asymmetric. There are many more players whose rating is below average than players whose rating is above it. Interesting fact is that the Elo histogram for chess looks exactly opposite - there are a lot more players with a high rating than players with a rating below average (Sismanis, 2010). The more detailed plot about players' rating distribution could be useful to a draw more reliable conclusion, since the histogram will not be able to reveal the reasons of such behavior. On the other hand, one of the main flaws of the current system of Bridge Federations is that it not only rates performance, but also frequency. It means that the more a player plays games, the more likely he is to have a high rate. It is possible to present both of these cases in one plot - namely how rating spread depends on the number of games. Output can be seen in Figure 5.9. The Figure shows that the model seems to not count the frequency, which was expected. What can be seen is that the less number of games, the less spread within ratings. It is natural, because some of them are out of reach. The more games are played, the wider the range is. Also

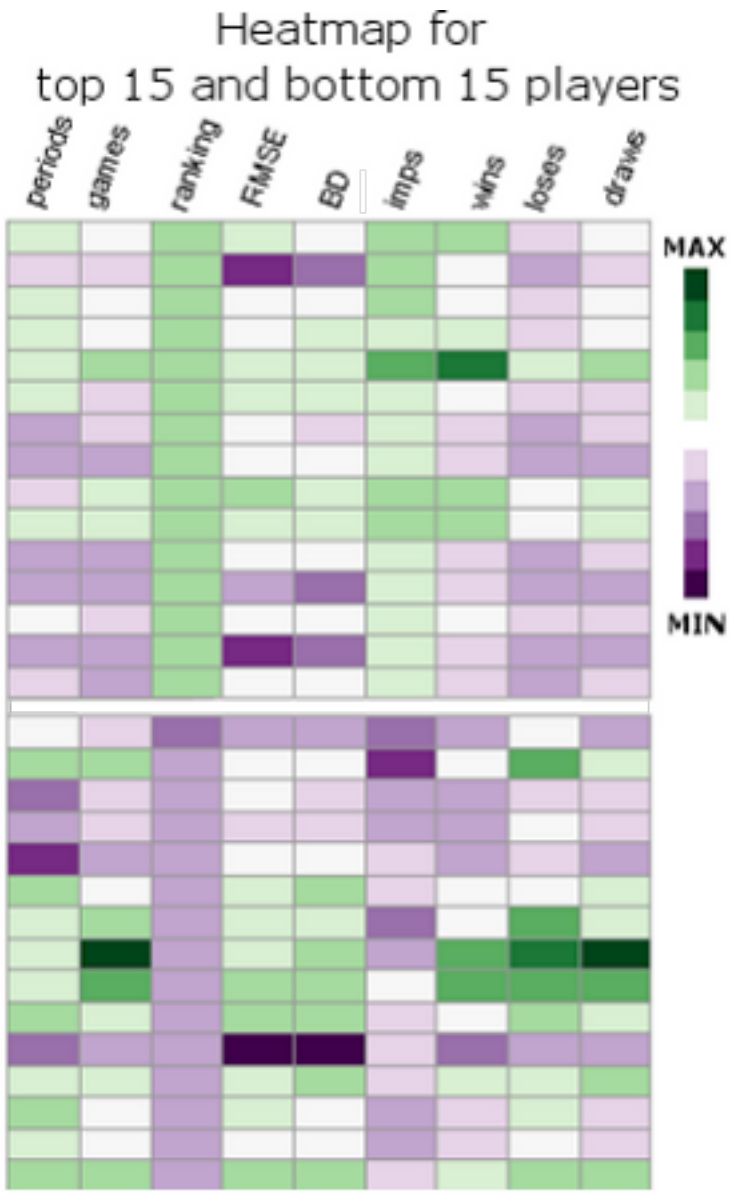

Figure 5.7: Figure shows a heatmap that visualizes 15 best and 15 worst players, order from the best to the worst. Top 15 are separated from bottom 15 with white space. All values have been scaled by corresponding column. The dark green color represents the maximum value (within the column), the white neutral and dark violet the minimal one.

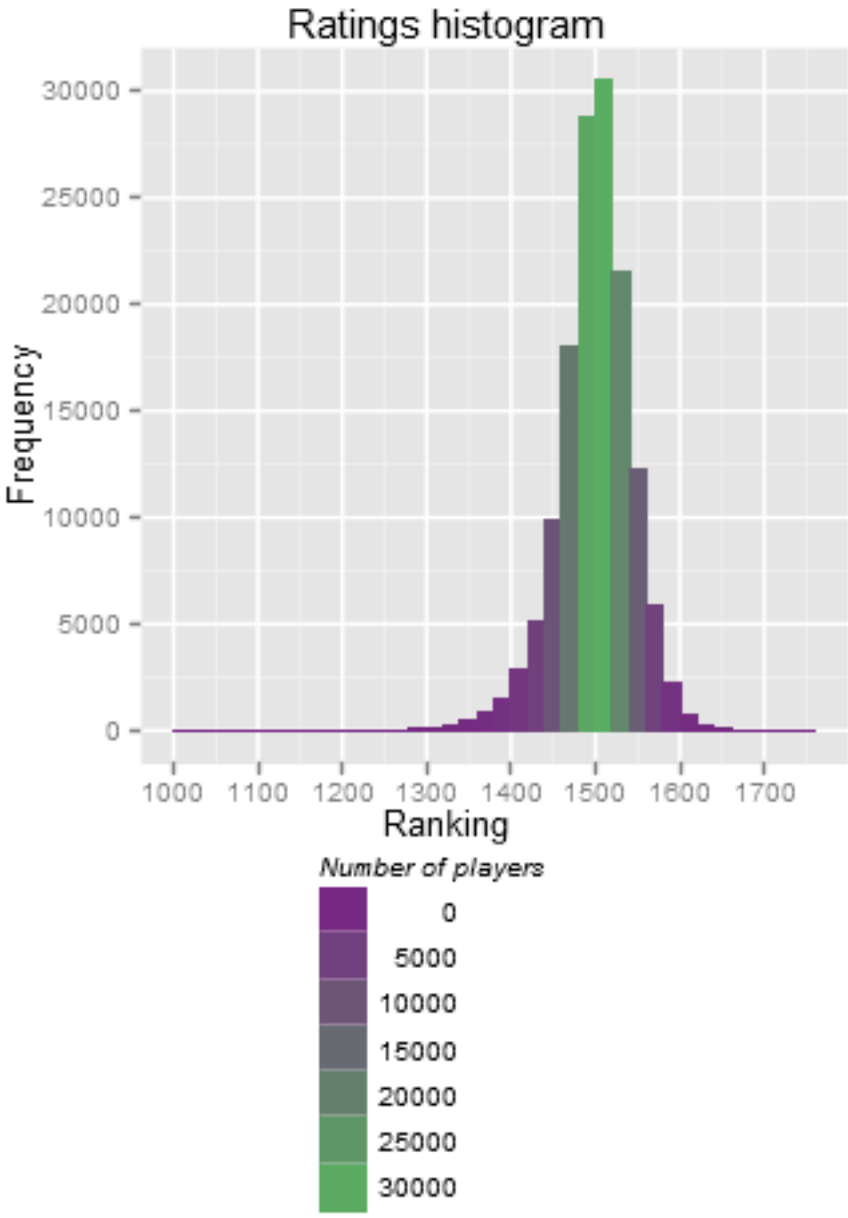

Figure 5.8: The histogram shows the rating distribution of players who have played at least 70 games. One can see that it is a little bit asymmetric. The left tail is heavier than the right one.

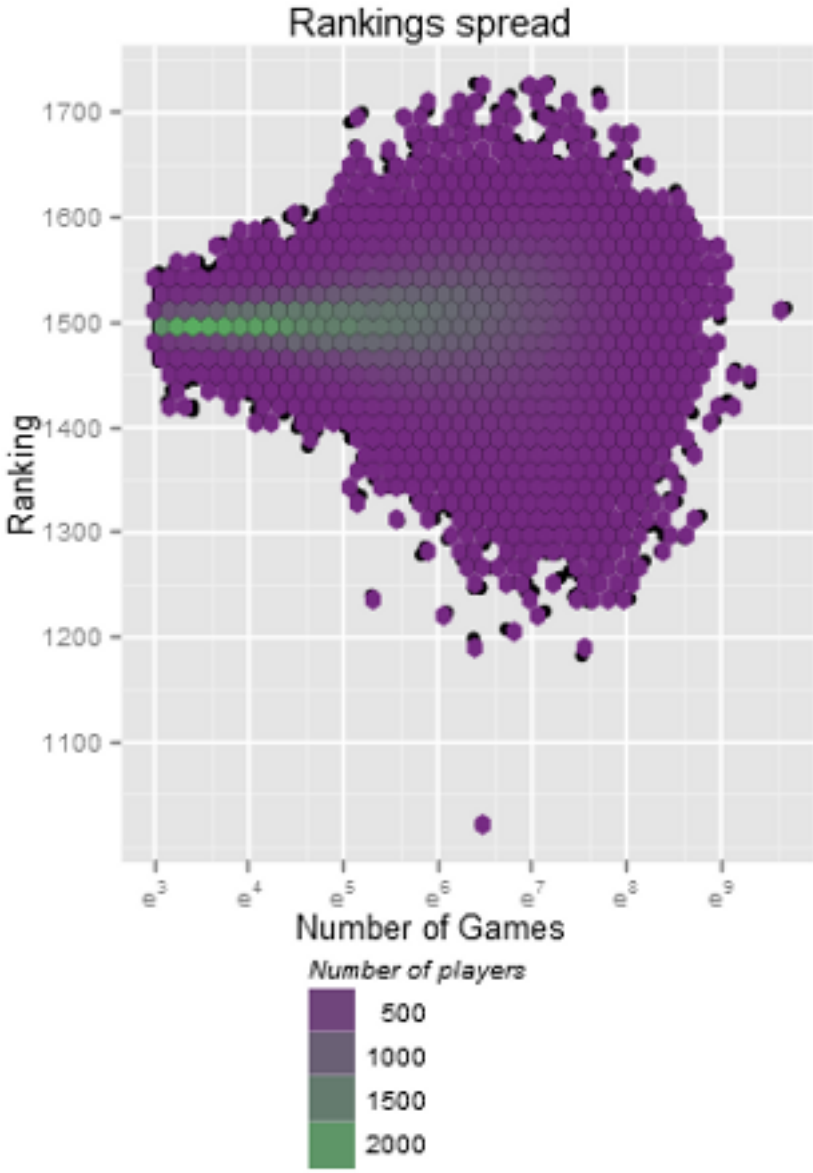

Figure 5.9: Figure shows the plot between log_e of number of games at x axis and the corresponding rating at y. Color indicates how often each rating occurred for a certain number of games. An important observation is that the number of games determines only how wide the range of ratings is, but it does not look like it is correlated with rating.

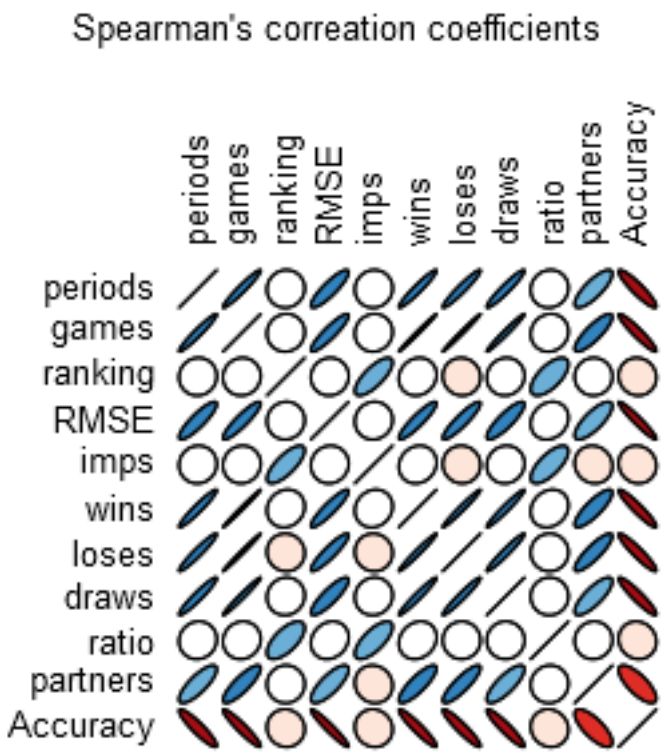

Figure 5.10: Visualization of Spearman correlation coefficients for player's statistics. The blue ellipses indicate positive value, while those with a red hue being negative. The more an ellipse looks like a circle, the closer value it is to 0.

the asymmetry can be seen quite well. One can see that most of the players have rating around $y = 1500$, however there are more outliers who have a lower rating than the histogram showed. The reason for this remains unclear for now, however further visualizations should provide more input to be able to explain this behavior.

Interesting insight provides the correlation matrix, hence it is shown in Figure 5.10. It is similar to the one created in the previous subsection. Once again, the Spearman's correlation coefficient has been used. Many interesting things can be spotted. First of all, there is a final proof that the model does not depend on frequency - the correlation coefficient between rating and the number of games/periods played is close to 0. The rating is definitely correlated with the

amount of imps and the win ratio. This is a very desired behavior, as described during the discussion about heatmap. Also, the assumption drawn at the beginning that there is some pattern about number of loses and ratings, seems to be consistent with the correlation coefficient - there seems to be a negative correlation between these two. At the first glance it might indicate that awarding and penalizing the player is not even. However, a closer look at the correlation matrix gives a more reliable answer - there is a correlation between loses and imps. Since rating is correlated with imps and imps are correlated with loses, it seems natural that rating and loses are also correlated. It means, that it is not necessarily caused by a flaw in the model - it is simply the nature of the data. However, the first reasoning that the players are too heavy penalized could be an explanation explanation for the histogram being asymmetric. A clear flaw can be spotted when looking at the correlation between accuracy and the number of different partners. The more partners, the less accurate the model is. This indicates, that λ needs re-considering and modification.

The last series of visualizations is about time series and how players statistics vary over time. Three players have been chosen who played in all periods. The first plot shown in Figure 5.11 would present how the rating was changing overtime. There are a few interesting observations about this plot. The main impression is that the ratings are relatively stable. For each player, there is one general trend that seems to continue. The green player has definitely the least amount of noise - he tends to progress in each rating period with a few minor exceptions. On the other hand, the red player seems to have a lot of big jumps - both on plus and on minus. The biggest one is between 13th March and 16th March. The blue player was constantly loosing rating half of the time, after which he had a small regular progress for a few rating periods and then he started to loose the rating again. To find the reason of the jumps of these two players, two additional time series have been created: one with IMPs and the second one with win ratio. They are presented in Figure 5.12 and Figure 5.13.

After taking a look at the number of imps there is no doubt that the model definitely puts a low weight to numbers of imps scored during the period. Considering the red player, who is actually known world champion, one can observe that he has a very high imp score after each period. His amount of imps differs a lot, but it is impossible to keep winning a lot of imps. What is the most important, is that looking at the imps time series, there is no real evidence that his skill was overrated. What the model seems to do, is that it expects a much higher win/lose ratio from him and hence it penalize him, even though his score is good. On the other hand, the green player scores only a little above 0. Actually, the blue player gains more imps than him. Looking at the win ratio time series clarifies the ratings of the players. The green one - who has the highest rating, has a comparable win ratio with the green player, however there is no doubt that he is worse in this metric as well. The blue one clearly is worse from

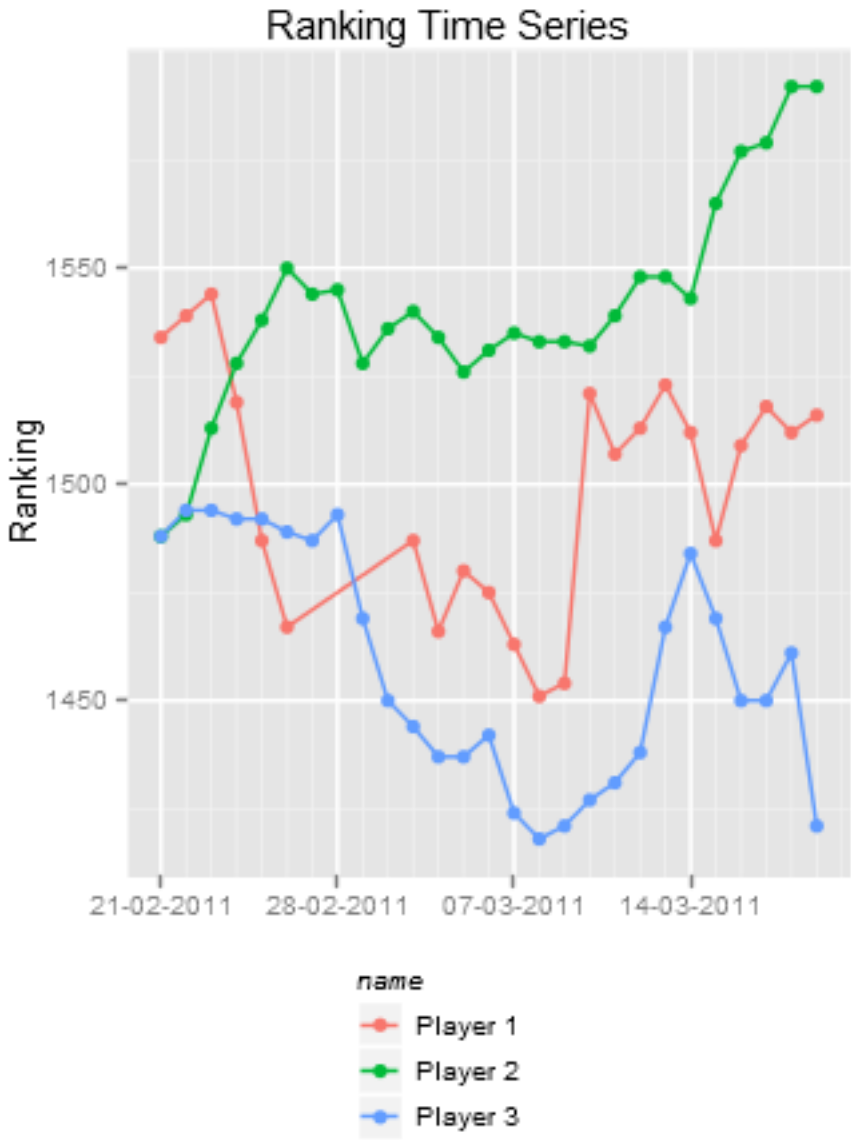

Figure 5.11: The figure shows how the rating has been changing over time for few chosen players.

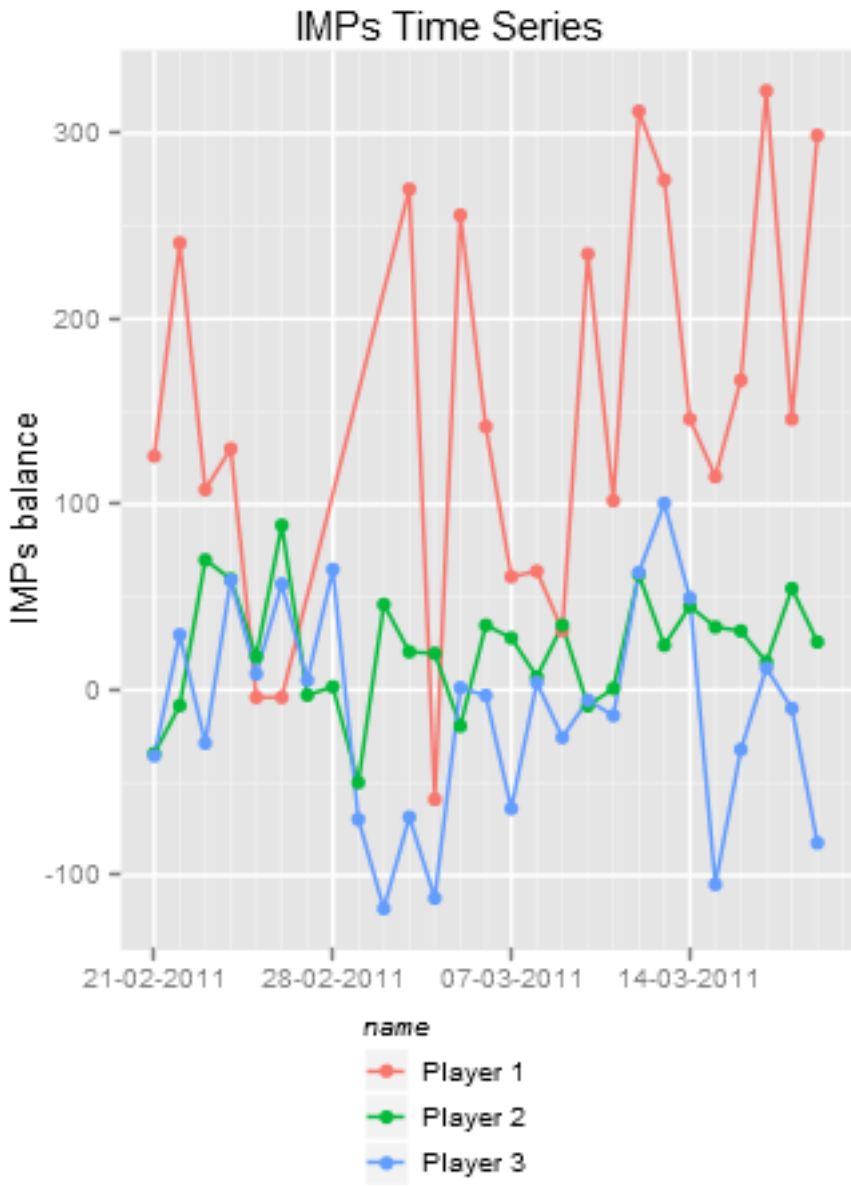

Figure 5.12: The figure shows the total IMPs balance after each rating period for three players whose rating have been presented in Figure 5.11. The important conclusion is that the ratings do not really reflect the total amount of imps gained.

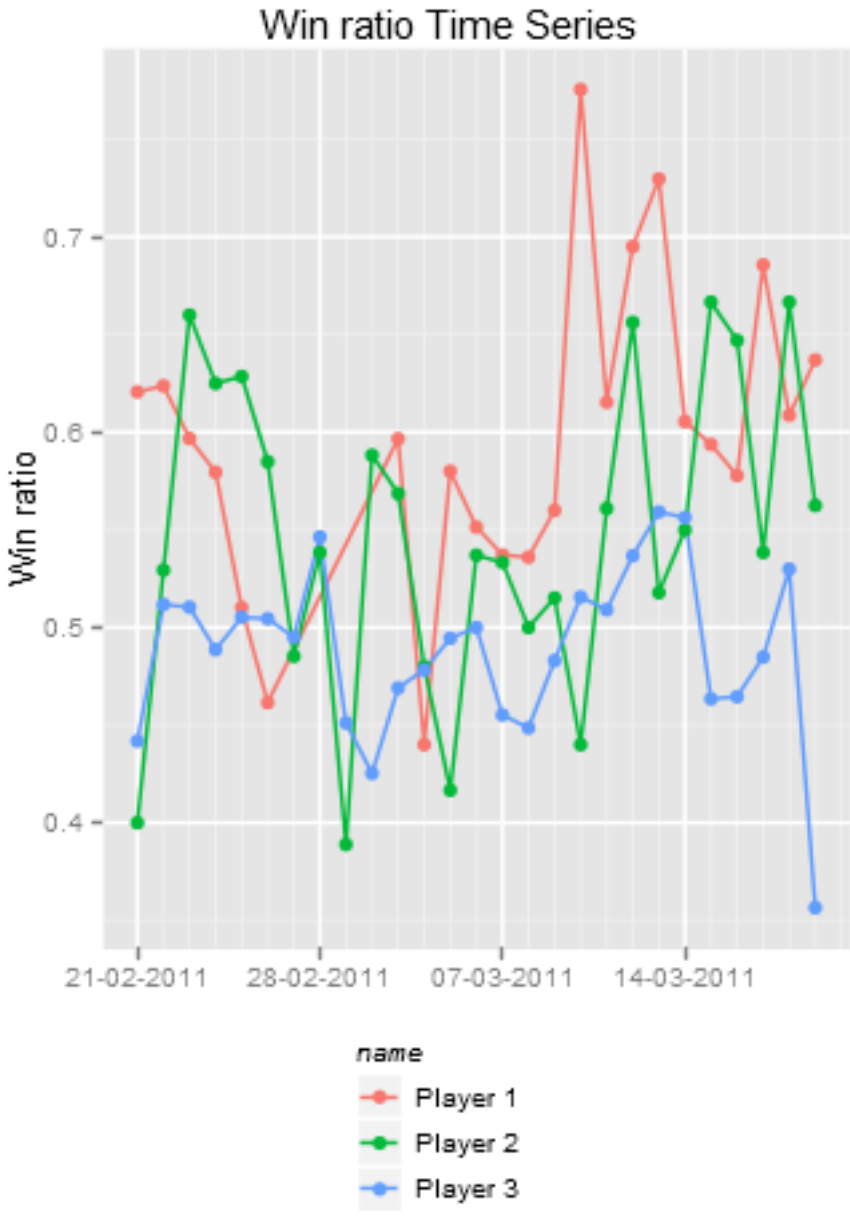

Figure 5.13: The figure shows how the win ratio changes over time for the same three players. After comparing this time series with the previous two presented in Figure 5.11 and Figure 5.12, one can observe a tendency to penalize a player even if his win ratio is really impressive.

both of them, what justifies his low position- he cannot get a high rating with a very poor win ratio. The very interesting fact remains, that the red player, even though he leads in both of the most important statistics - the ones that are correlated with the rating - is only a little below average. It seems that he is penalized too much for not keeping an extremely good shape, which results in treating him as an average player. This bias could be corrected by assigning much greater weight to the total imps statistic and less to the win ratio.

5.4 Visualization Summary

A number of different visualization techniques have been used in order to verify and analyze the results of the model. From the first part of the visualization, the most important conclusion was that there is an extremely big correlation between the number of games played and RMSE and the prediction accuracy. For each period about 50% players were outliers and their expected score was matching the real one from 100% to around 40%, with some additional outliers in 30% and 20% as well. It also showed that there is a little negative correlation between rating period and accuracy, which is also not desired behavior.
The second part was more optimistic. It was proven that there is no correlation between statistics that should not affect rating - most importantly, the rating was not correlated with the number of games. On the other hand, the expected correlations exist - between rating and win ratio and imps. This is desired, because these statistics characterize a good player. The correlation coefficient also showed an issue with estimating the combined strength of partners. On the other hand, the time series revealed a problem with weighting imps and win ratio. Even though both are important, the first one is generally considered more reliable in bridge. As for now, the win ratio has greater influence on the final ratings.

Chapter 6

Discussion

The main goals of the thesis were:

- Acquire real life bridge statistics
- Model bridge player's performance
- Visualize the results

All of the goals have been addressed by the thesis. The statistics of real life players have been acquired by web-crawling freely accessible services. The process was extremely time-consuming, however relatively simple to carry out. After filtering the noise and erroneous or incomplete data, the total number of players was $203,278$ who played in total $21,684,154$ games for $2,226,279$ deals that took place during the period of Feb 21st to 13th of May - 82 days. The acquired, parsed and filtered data has been used to achieve two remaining goals: To verify the invented model of a bridge player's performance and to measure its reliability, optimizing it and finally visualize the results.

The summary of the modeling part is described in section 6.1 and a summary of the visualization in section 6.2. Section 6.3 is ending this thesis and contains a description of planned future work.

6.1 Summary of the Model

The first goal has been achieved by extending the Elo rating system, which has been created for two-player games like chess. It models the performance of each player as a random variable following Logarithmic Distribution. Since two pairs always participate in bridge, they are mapped to the two-player scenario by averaging ratings of players within the partnerships. To enhance predictions, the base model has been extended by four new parameters. The final version of the model is given with the formula:

$$R_n = R_{n-1} + K_d * (S - \frac{1}{1 + 10^{\frac{((r_{o1}+r_{o2})/2+\lambda_o+\rho_o)-((r_{p_1}+r_{p_2})/2+\lambda_p+\rho_p)+\gamma}{400}}}) \quad (6.1)$$

Where

- $(r_{o1} + r_{o2})/2$ - Is the average of opponents' ratings
- $(r_{p_1} + r_{p_2})/2$ - Is the average of players' ratings, for who the computations take place
- γ - Is the adjustment based on the skill level of all players at other tables
- λ - Is the adjustment for the rating of a pair basing on how many number of games two players played together
- ρ - Is the adjustment of the rating difference between partners
- K_d - Is modified K-factor based on the obtained scores during whole rating period

In addition, the unreliability of close wins has been referred by extending considering a game draw if the score is between $-1 \leq s \leq 1$.

To be able to asset and compare different versions of a model, two metrics have been used: Root Mean Square Error and Binomial Deviance. The final accuracy, that have been counted on the basis of matches played by all non-provisional players[1] is $50,018\%$. It means that the predictions are slightly better than the null system, which always sets probability of winning for both players to 50%. The results of the model might not look impressive. However, any improvement

[1] Players who played more than 10 games

of the random system has been considered as a success. One should observe that chess, for which the basic Elo system has an accuracy of 55%, is a two-players game and the only additional factor is who had the first move. Bridge, on the other hand, is not only a partnership game, as the scores are calculated by comparing results of many independent tables. This introduces an incredible amount of noise and a lot of additional questions and greatly complicates the problem, which resulted in poor - but still better than random - predictions. One can also realize that if the task was easy, most probably some variation of Elo rating system would be applied to bridge already, as it was for many other systems.

6.2 Summary of Visualization

The last part of the thesis - visualization - discovered a few interesting features about the data. First of all it showed that the more players play, the less accurate the system is. Such conclusion has been reached by plotting accuracy in many different ways, including for each period, for each player, and for each player in each period. The visualized matrix of Spearman's correlation coefficients confirmed that fact. Also, there is a lot of noise for predicting scores - according to boxplot, for each period there is incredibly wide range of accuracy - from 100% to even 30%. However at least half of all players in each period had about 50% accuracy. The most likely reason is not that effective predictions and not enough quality data. Even though there are a lot of games played, there are also a lot of different players, some of them play only few periods, not necessary regularly. This makes it hard to make reliable measurements between periods. Secondly, it was showed that the histogram of rating distribution between players is asymmetric. Its left tail is heavier than the right one, which means that there are more players below an average rating than players above. It is not clear what the reason for this is, however one very likely idea is that players are too heavily penalized. This seems to be confirmed by the conclusion reached after visualizing time series for three parameters for three players. The statistics that were used for drawing them were: rating, imps and win ratio. It is clear why the rating has been chosen - after all it is the player's performance. The second two were proved to be correlated to it, using Spearman's correlation coefficient. The results of the analysis of these three graphs were that a player who scored the biggest amount of imps and had the best win ratio has been rated by the system only a little above average. The reason is that a bigger weight is assigned to the win ratio, while it is imps which is the more reliable metric. Bridge is a card game, which means it is hard to predict a winner during a short period. However, good players will very often have a lot of imps on plus, while their win ratio can vary a lot. Even if it is reasonable high, the system stills penalize for

underperformance.
Another observation was that opposite to the current rating system used officially by many federations, the obtained model measures only bridge players performance and not the frequency by which players play. Such conclusion could be made based on the 3D plot of ratings and the number of periods played. Again, the Spearman's correlation coefficient was used to ensure a proper conclusion.

6.3 Future Work

The obtained model has been proved to perform better for the given data set than the null system. However, the visualization process revealed many flaws, which should be corrected. The most crucial one was the unclear convergence of the rating system. Proving such property is probably the most important goal. The second one is to enhance predictions and make them more accurate. A first step towards achieving this goal could probably be to explicitly model teams and draws. These features are already implemented by TrueSkill (Ralf Herbrich and Graepel, 2007). It would allow to analyze the pairs at other tables in a much more generic way than what is done now. Another one is to assign lower weight to the win ratio and higher to imps scored. Next improvement is to not only count the number of games with partner, but also consider when they played together last time. It was not that important for the current data set, since it contained statistics for only 82 days. However it is definitely required for a serious system which is supposed to replace a Masterpoint system used by Bridge Federations. Last, but not least, it would be important to apply more scientific methods to optimize parameters, for example stochastic gradient descent technique (Spall, 2003).
To sum up, the following list of the most important tasks for future work has been defined:

- Migrate from Elo to TrueSkill to calculate players combined strength more accurate and to define draw in more formal way
- Lower the weight for win ratio and increase importance of IMPs
- Take into account not only how many games partnership played together, but also when it was
- Use more advanced methods for parameters optimizations

Bibliography

Allen, C. and Appelcline, S. (2006). Collective Choice: Competitive Ranking Systems. [online].

Bolboaca, S. D. and Jantschi, L. (2006). Pearson versus Spearman, Kendall's Tau Correlation Analysis on Structure-Activity Relationships of Biologic Active Compounds. *Leonardo Journal of Sciences*, 5:179–200.

Chebotarev, P. Y. and Shamis, E. (2006). Preference fusion when the number of alternatives exceeds two: indirect scoring procedures. Technical report, Institute of Control Sciences of the Russian Academy of Sciences.

Curley, J. (2010). [online].

EuropeanGoFederation (2011). Egf official ratings system.

Fry, B. (2008). *Visualizing data*. O'Reilly Series. O'Reilly Media, Inc.

Glickman, M. E. (1995). A comprehensive guide to chess ratings.

Glickman, M. E. (2001). The Glicko system. [online].

Glickman, M. E. and Jones, A. (1999). Rating the chess rating system. *Chance*, 12.

InternationalChessFederation (2011). The working of the fide rating system.

James Piette, L. P. and Anand, S. (2011). Evaluating Basketball Player Performance via Statistical Network Modeling. In *MIT Sloan Sports Analytics Conference 2011*.

Janert, P. (2010). *Data Analysis with Open Source Tools*. O'Reilly Series. O'Reilly Media.

Kschischang, F. R., Frey, B. J., and Loeliger, H. A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

Miller, S. C. (2003). The uscf elo rating system.

Moser, J. (2011). The Math Behind TrueSkill.

Park, J. and Newman, M. E. J. (2005). A network-based ranking system for US college football. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(10):P10014+.

Pham, L., Christadore, L., Schaus, S., and Kolaczyk, E. D. (2011). Network-based prediction for sources of transcriptional dysregulation using latent pathway identification analysis. *Proceedings of the National Academy of Sciences*, 108(32):13347–13352.

Ralf Herbrich, T. M. and Graepel, T. (2007). TrueSkill: A Bayesian Skill Rating System. In *Advances in Neural Information Processing Systems 20*, pages 569–576. MIT Press.

Runyan, B. (1997). The World Football Elo Rating System. [online].

Segaran, T. (2007). *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly, Beijing.

Sismanis, Y. (2010). How I won the Chess Ratings - Elo vs the Rest of the World. *CoRR*, abs/1012.4571.

Smith, M. R. (2006). Modeling the performance of a baseball player. Offensive production. Master's thesis, Brigham Young University.

Sonas, J. (2011). Deloitte/fide chess rating challenge.

Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc.

Weng, R. C. and Lin, C.-J. (2011). A Bayesian Approximation Method for Online Ranking. *Journal of Machine Learning Research 12 (2011)*, pages 267–300.

Wickham, H. (2006). An introduction to ggplot: An implementation of the grammar of graphics in R.