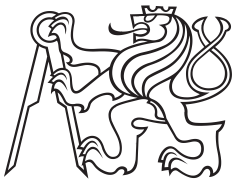
Master Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Searching for betting market inefficiencies with mathematical programming

Zdeněk Syrový

Supervisor: Ing. Gustav Šír, Ph.D.
May 2023

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Syrový Zdeněk** Personal ID number: **474404**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Data Science**

II. Master’s thesis details

Master’s thesis title in English:

Searching for betting market inefficiencies with mathematical programming

Master’s thesis title in Czech:

Hledání neefektivit na sázkařských trzích s pomocí matematického programovaní

Guidelines:

Sports betting provides an excellent playground for statistical hypotheses testing of efficiency of the individual markets, which has been exploited in a number of works [1,2,3]. However, the interdependent structure of the individual markets (win-draw-loss, asian handicapping, etc.) for a given match provides an interesting opportunity to explore their inefficiencies jointly. A simple example of such joint inefficiency is arbitrage investment. The subject of this work is to explore a wider range of such joint inefficiencies by encoding the underlying market relationships in the language of mathematical programming.
1) Review existing literature on betting market efficiencies with a special focus on arbitrages and their extrapolations to multiple markets.
2) Research existing online data sources for predictive sports markets (bookmakers, tipsters, exchanges), with a special focus on cross-market data.
3) Obtain a significantly large and suitable dataset for the analysis, preprocess and validate the data.
4) Find a suitable encoding of the task as a mathematical programming problem.
5) Experiment with different settings, definitions of inefficiencies, and their relaxations.
6) Analyze and discuss your results.

Bibliography / sources:

[1] Hubáček, Ondřej, Gustav Šourek, and Filip Železný. 'Exploiting sports-betting market using machine learning.' International Journal of Forecasting 35.2 (2019): 783-796.
[2] Matej, Uhrín, et al. 'Optimal sports betting strategies in practice: an experimental review.' IMA Journal of Management Mathematics (2021).
[3] Angelini, Giovanni, and Luca De Angelis. 'Efficiency of online football betting markets.' International Journal of Forecasting 35.2 (2019): 712-721.

Name and workplace of master's thesis supervisor:

Ing. Gustav Šír, Ph.D. Intelligent Data Analysis FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **15.09.2022** Deadline for master's thesis submission: **10.01.2023**

Assignment valid until: **19.02.2024**

Ing. Gustav Šír, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I want to thank my supervisor Gustav Šír for his endless patience, positive mindset, and a lot of good ideas.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in preparation for university theses.

In Prague, 25. May 2023

Abstract

This thesis investigates the identification and exploitation of betting market inefficiencies through the application of mathematical programming techniques. The efficiency of betting markets has long been a topic of interest, as they represent a platform where participants can wager on uncertain outcomes, such as sporting events. The hypothesis underlying this research is that there exist systematic biases or inefficiencies in these markets that can be detected and leveraged for profitable opportunities.

The study focuses on the use of mathematical programming, specifically linear programming and integer linear programming, to model and optimize betting strategies for arbitrage betting by formulating mathematical models that incorporate various factors such as odds, probabilities, market imbalances, and historical data.

Keywords: Betting markets, market inefficiencies, mathematical programming, linear programming, integer linear programming, optimization, betting strategies, computational complexity

Supervisor: Ing. Gustav Šír, Ph.D.

Abstrakt

Tato práce se zabývá identifikací a využíváním neefektivity sázkového trhu pomocí technik matematického programování. Efektivita sázkových trhů je již dlouho předmětem zájmu, protože představují platformu, kde mohou účastníci sázet na nejisté výsledky, například sportovních událostí. Hypotéza, z níž tento výzkum vychází, je, že na těchto trzích existují systematické chyby nebo neefektivnosti, které lze odhalit a využít k vytváření ziskových příležitostí.

Studie se zaměřuje na využití matematického programování, konkrétně lineárního programování a celočíselného lineárního programování, k modelování a optimalizaci sázkových strategií pro arbitrážní sázení. Pomocí formulace matematických modelů, které zahrnují různé faktory, jako jsou kurzy, pravděpodobnosti, nerovnováha trhu a historická data.

Klíčová slova: Sázkové trhy, neefektivita trhu, matematické programování, lineární programování, celočíselné lineární programování, optimalizace, sázkové strategie, výpočetní složitost

Contents

Figures

Tables

Chapter 1

Introduction

Sports betting has long been a popular pastime for millions of people worldwide. However, despite the widespread participation, the betting market is a complex system, with dynamic interactions among the various participants. Betting markets are often perceived as being efficient, with prices reflecting all available information, making it difficult for bettors to outperform the market consistently.

In this thesis, we explore the possibility of identifying market inefficiencies in football betting using mathematical programming techniques. To provide a solid foundation for this exploration, we first define basic terms.

Once these terms have been defined, we review the existing literature on football betting market inefficiencies and the mathematical approaches that have been used to identify them. We then present a novel mathematical programming model designed specifically for identifying market inefficiencies in football betting.

We test the model on a large data set of historical betting odds and outcomes from football matches. The results are evaluated using various statistical measures, including profit and loss, and the ability to identify situations in which the odds offered by bookmakers do not accurately reflect the underlying probabilities of outcomes.

Overall, our thesis aims to contribute to the growing body of research on football betting markets and provide insights into how mathematical

programming can be used to identify and exploit market anomalies in football betting.

1.1 Related terms

1.1.1 Bettor and Bookmaker

A *bookmaker* is a person or company that takes bets on sporting events and sets odds on the outcome of those events. The bookmaker's goal is to make a profit by ensuring that the amount of money wagered on each outcome is balanced so that regardless of the outcome, they will make a profit.

Soft and Sharp bookmakers

In general, bookmakers can be divided into two non-overlapping categories sharp and soft bookmakers.

Soft and sharp bookmakers differ significantly in terms of their approach to setting odds and managing their customer base. Soft bookmakers typically target casual bettors and aim to attract them with attractive odds and promotions. They achieve this by setting odds that may not accurately reflect the true probabilities of an outcome and by having a larger margin on their odds, allowing them to generate a profit even if their customers win a significant portion of their bets. However, soft bookmakers often have lower limits on bets and are more likely to limit or ban winning players.

Sharp bookmakers, on the other hand, focus on catering to professional bettors by setting more accurate odds and offering higher limits on bets. They often have a smaller margin on their odds and are less likely to limit or ban winning players, as they recognize the value that professional bettors can bring to their business. Sharp bookmakers use advanced risk management strategies to manage their exposure to potential losses and are more likely to have a long-term focus on profitability.

The difference between the two is also part of the ongoing research such as in [HW23b] or [HW23a].

A *bettor*, on the other hand, is a person who places a bet with a bookmaker, hoping to win money by correctly predicting the outcome of an event.

1.1.2 Odds

Bookmaker *odds* are numerical expressions of the likelihood of a particular outcome in a sports event. Bookmakers use their knowledge, expertise, and a variety of factors to calculate odds for different outcomes of an event. The odds provided by bookmakers are used by bettors to place bets on various outcomes. *Implied probabilities* are the inverse of odds. Even though we use the term probability here, *implied probabilities* are not probabilities in strict mathematical meaning as their sum is usually larger than one because of the bookmaker margin.

Bookmaker's margin

The bookmaker's margin is the amount by which the total implied probability of all possible outcomes in an event exceeds 1. It represents the bookmaker's profit margin. This margin gives the bookmaker edge thus making the game not fair for the better. The margin can be calculated in the following way.

Let n be the number of possible outcomes of an event, and let o_i be the odds on outcome i. Then the bookmaker's odds margin M is:

$$M = \sum_{i=1}^{n} \frac{1}{o_i} - 1 \tag{1.1}$$

There are also multiple types of margins as stated in [CLP03]. The one shown in Equation 1.1 is called uniform.

1.2 General betting strategies

In general, there are two basic betting strategies used by bettors value betting and arbitrage betting.

1.2.1 Value betting

Value betting is a strategic approach used in sports betting and other forms of gambling, in which a bettor places a wager based on the belief that the odds offered by a bookmaker or betting exchange are more favorable than the true probability of an outcome occurring. The goal of value betting is to identify situations where the odds are mispriced, such that the expected value of the bet is positive in the long run. One of the typical approaches in value betting is implementing machine learning algorithms and statistical models to analyze historical data and identify patterns that can be used to make predictions about future events. The success of a value betting strategy depends on the accuracy of these predictions and the ability to effectively manage risk and bankroll, which is a theme of [Buc03]. Another approach is to exploit soft bookmakers' worse estimate of probabilities by confronting them with sharp bookmakers' odds.

1.2.2 Arbitrage betting

Arbitrage betting is a technique used by bettors to exploit discrepancies in odds across multiple bookmakers in order to guarantee a profit regardless of the outcome of the event. The idea behind arbitrage betting is to identify situations where the odds for all possible outcomes of an event are such that a profit can be made by placing bets with different bookmakers, each offering different odds. By carefully calculating the appropriate stakes to place on each bet, the bettor can lock in a profit, regardless of the outcome of the event. In this thesis, we focus on this particular technique.

1.3 Related work

As far as we know, no authors tried using mathematical programming to find arbitrage opportunities. However, a lot of research work was done in the domain of creating prediction models and searching for betting market inefficiencies.

In the domain of betting market efficiency, for example, authors in [DF8] looked at market inefficiencies in German football at the start of the season. Authors in [GT91] tested the hypothesis of whether the NFL(American football) market is efficient and were able to find biases, for example, underdog bias. Favorite-longshot bias was also found by [WW94] in the baseball league. And many other works were done in this domain with different results.

In the domain of predictive models, for example, authors in [H19] used convolutional neural networks for match outcome prediction. In contrast, authors in [CDLR02] used a simpler independent Poisson model for predicting 1x2 market results. And authors in [HA10] used an ELO system similar to one used in chess for match predictions.

There is also some work done in comparison of different approaches such as in [MGOF21] where authors look at sports betting as an investment and use economic approaches such as applying modern portfolio theory and applying Fractional Kelly Criterion for optimal fund division or in [H21] authors experimentally tested multiple sport-predicting methods on football historical data.

On arbitrage betting some research work was done as well in [FVN13] authors successfully found arbitrage betting opportunities and implied relationship between market structure and market efficiency. Authors in [VDM09] tried several betting models including arbitrage and suggested inconsistencies in the betting market using football data. Arbitrage betting for horse racing was suggested by [HZ90]. And in [FNV09] authors tried inter-market arbitrage betting using several bookmakers and exchange sites.

1.4 Background

This section serves as an overview of mathematical programming, that will be used later in the thesis.

Mathematical programming refers to the process of formulating and optimizing mathematical models to solve complex decision-making problems. It involves the use of mathematical techniques, such as optimization and linear algebra, to find the best possible solution within a given set of constraints. Two types of mathematical programming that will be used in this theses are linear programming (LP) and integer linear programming (ILP). Mathematical programming has many interesting real applications such as in [BFGM00], where authors used mathematical programming for classical scheduling problems.

Linear Programming (LP). Linear programming is a mathematical optimization technique used to solve problems with linear objective functions and linear constraints. LP assumes that the relationships between variables are linear, and the objective is to find the optimal values of the decision variables that maximize or minimize the objective function, subject to the given constraints. The decision variables are typically non-negative and continuous in LP. On solving linear programs, a lot of research was done such as in [Chv83] or [Dan02] by two of the most prominent LP and ILP researchers.

Integer Linear Programming (ILP). Integer linear programming extends the concepts of LP by introducing an additional requirement that the decision variables can take on integer values. This restriction makes ILP problems more challenging and computationally complex compared to LP. The presence of integer variables allows for modeling discrete decision choices, such as selecting or assigning whole numbers of items or activities. Integer linear programming has also a lot of applications as shown for example in [Aba89] where authors used ILP for fleet assignment problems.

Solving ILP problems often requires specialized algorithms, such as branch and bound or cutting plane methods, which explore different feasible solutions and progressively narrow down the search space to find the optimal integer solution. These methods may involve solving a series of LP relaxations, where the integer constraints are relaxed to obtain a continuous solution that serves as a lower bound for the optimal integer solution. In integer linear programming also a lot of research work was such as in [CKS90],

where authors introduced the cutting planes method of solving integer linear programs.

Chapter 2

Arbitrage betting

We have already defined Arbitrage betting in the previous chapter. Here, we will go into more detail. Firstly we need to define all the possible markets that bookmakers offer.

2.0.1 1x2 market

The most common market is called the 1x2 market, and in football, there are three possible outcomes that bettor can bet on. Win of the home team (also called 1), draw (x), and away team wins (2). Table 2.1 shows the results of the bets based on how many goals each team scored.

		Home Goals				
		0	1	2	3	4
Away Goals	0	Draw	Home	Home	Home	Home
	1	Away	Draw	Home	Home	Home
	2	Away	Away	Draw	Home	Home
	3	Away	Away	Away	Draw	Home
	4	Away	Away	Away	Away	Draw

Table 2.1: Indicates to which result in the box belongs

2.0.2 Both to score

Another market is both to score market. The bet here is whether both teams score a goal or not. Table 2.2 shows possible outcomes.

		Home Goals				
		0	1	2	3	4
Away Goals	0	No	No	No	No	No
	1	No	Yes	Yes	Yes	Yes
	2	No	Yes	Yes	Yes	Yes
	3	No	Yes	Yes	Yes	Yes
	4	No	Yes	Yes	Yes	Yes

Table 2.2: Indicates to which result in the box belongs

2.0.3 Asian Handicap

Asian Handicap is a type of market that creates a form of handicap betting that aims to level the playing field between two teams of different abilities. In this type of betting, a handicap is applied to one of the teams in a way that one of the teams starts with a points advantage.

The most simple form of handicap is the point five handicap market. In this market, handicaps take the form of .5 numbers, which allows for easy calculation of results, the handicap is added or subtracted from the match outcome, and the winner is based on the adjusted scoreline. An example of +1.5 home handicap can be seen in Table 2.3.

Another form of handicap is whole point handicap. In this case, whole points are added or subtracted, and if the result is a draw, then stakes are refunded. An example is in Table 2.4.

The last type is the middle handicap, which is basically a combination of point five and whole points handicaps. Handicaps take form x.25 or x.75. Half of the stake goes to the whole point handicap and the other half to point five handicaps. Both of these are then evaluated independently of each other, which creates new outcomes as can be seen in Table 2.5, where in case the match ends in a score 0:1, then point five part of the bet is lost(0.5:1) and whole point part is refunded(1:1). in case of x.25 new outcome is a half win and half refund.

		Home Goals				
		0	1	2	3	4
Away Goals	0	Home	Home	Home	Home	Home
	1	Home	Home	Home	Home	Home
	2	Away	Home	Home	Home	Home
	3	Away	Away	Home	Home	Home
	4	Away	Away	Away	Home	Home

Table 2.3: +1.5 home or -1.5 away Asian Handicap

		Home Goals				
		0	1	2	3	4
Away Goals	0	Home	Home	Home	Home	Home
	1	Return	Home	Home	Home	Home
	2	Away	Return	Home	Home	Home
	3	Away	Away	Return	Home	Home
	4	Away	Away	Away	Return	Home

Table 2.4: +1 Home Asian Handicap

		Home Goals			
		0	1	2	3
Away Goals	0	Home	Home	Home	Home
	1	1/2 lose 1/2 return	Home	Home	Home
	2	Away	1/2 lose 1/2 return	Home	Home
	3	Away	Away	1/2 lose 1/2 return	Home
	4	Away	Away	Away	1/2 lose 1/2 return

Table 2.5: +0.75 Home Asian Handicap

2.0.4 Over-Under

The over-under is a type of market where is betted on whether the sum of goals exceeds a certain threshold. Same as in the case of Asian handicaps, there are point five and whole points types of over-under bets. Both are evaluated the same way as in Asian handicaps but with the total sum of goals. An example of point five can be seen in Table 2.6 and the whole point in table. 2.7

		Home Goals				
		0	1	2	3	4
Away Goals	0	Under	Under	Over	Over	Over
	1	Under	Over	Over	Over	Over
	2	Over	Over	Over	Over	Over
	3	Over	Over	Over	Over	Over
	4	Over	Over	Over	Over	Over

Table 2.6: Over/Under 1.5

		Home Goals				
		0	1	2	3	4
Away Goals	0	Under	Refund	Over	Over	Over
	1	Refund	Over	Over	Over	Over
	2	Over	Over	Over	Over	Over
	3	Over	Over	Over	Over	Over
	4	Over	Over	Over	Over	Over

Table 2.7: Over/Under 1

2.0.5 Draw no bet

Draw No Bet is often advertised as a standalone market, but it is simply Asian handicap +0, which means that if a match is drawn then stakes are refunded.

2.1 Simple model of arbitrage betting

Now that we have all for us relevant markets defined, we can move to some simple examples of arbitrage betting.

2.1.1 One market arbitrage betting

If only mutually exclusive(non-overlapping) markets are considered, the detection of arbitrage opportunities is straightforward. A Set of (non-overlapping)odds is an arbitrage opportunity if and only if their implied probabilities sum is lower than one. Once we detect the opportunity all that is left is to calculate how much to bet on each outcome. The most used

strategy is to divide the total stake in such a way as to ensure the same profit independently of the result of the match.

To calculate the optimal betting strategy we first calculate the optimal profit $profit^*$ given odds o total stake S, number of outcomes n, and bet on outcome i is o_i. Equation 2.1 gives optimal profit value. then we use the fact that the profit of a bet can be expressed as $profit = b_i o_i - S$. When we insert optimal profit $profit^*$ we can solve for b_i and get equation 2.3 which calculates bet b_i on outcome i to get optimal profit $profit^*$.

$$profit^* = \frac{S}{\sum_{i=1}^{n} p_i} - S \tag{2.1}$$

$$S = \sum_{i=1}^{n} b_i$$

$$profit^* = b_i o_i - S \tag{2.2}$$

$$b_i = (profit^* + S)/o_i \tag{2.3}$$

2.2 Max-min Mathematical programming model

Once we want to start using multiple markets with overlapping results, the relationships between different outcomes start to be more complicated, and different formalism is needed to find optimal solutions.

In our case, we chose integer linear programming:

$$\begin{array}{ll} \max & profit \\ \text{subject to} & Ax - profit \geq 0 \\ & profit \geq 1 \\ & x \geq minbet * y \\ & x \leq maxbet * y \\ & y \in \{0,1\}^n \\ & x \in [0,\infty)^n \\ & profit \in [1,\infty) \\ & A \in \mathbb{R}^{(points+1)^2 \times n} \end{array}$$

2.2.1 Variables of mathematical programming model

In this mixed integer linear program, there are two sets of variables x and y and $profit$ variable to be maximized. Vector variable x represents a betting vector and the length of the vector is n which is the number of market outcomes(i.e. for a 1x2 market it is 3). After the optimal value is found, vector x is our betting strategy. Binary vector variable y is also the number of market outcomes long and is set for one particular outcome to one if and only if the market outcome is used in the final betting strategy and to zero otherwise.

2.2.2 Constants of mathematical programming model

Constants in our model are matrix A are values $minbet$, $maxbet$, and $points$. As names suggest, $minbet$ and $maxbet$ represent the minimum and the maximum bet that can be bet on one outcome. By multiplying these constants by the binary variable y, constraints are active if $y = 1$ and inactive otherwise. For matrix A and value $points$, we need to go into more detail.

Construction of matrix A

Matrix A is constructed by creating one column of the matrix for each market outcome possible. Each row then represents one match outcome(e.g., 1:0) in the following way:

$$\text{A[r,c]} = \begin{cases} -1 & \text{if bet is lost in this particular match outcome} \\ o-1 & \text{if it is winning bet} \\ 0 & \text{if stakes are refunded} \\ \frac{o-1}{2} & \text{if outcome is half-win} \\ -\frac{1}{2} & \text{if outcome is half-lose} \end{cases}$$

In constraint, $Ax - profit \geq 0$ Ax is a vector of length $(points + 1)^2$, where $points$ is the maximum possible considered number of scored goals. $(points + 1)^2$ is then the number of all possible match outcomes ranging from 0:0 to points:points. A maximum number of goals is not capped in a football

game, so we need to set *points* constant high enough for the vast majority of games to fit in but not that high to keep good performance(the number of constraints is quadratic in *points*). An example of how matrix A can look is in Table 2.8, here only 1x2 market is present. Additional markets would be added just by appending extra columns. In the table, odds three are used for home, away, and even for match draw.

Match Outcome	Home	Draw	Away
0:0	-1	2	-1
1:0	2	-1	-1
2:0	2	-1	-1
0:1	-1	-1	2
1:1	-1	2	-1
2:1	2	-1	-1
0:2	-1	-1	2
1:2	-1	-1	2
2:2	-1	2	-1

Table 2.8: Example of matrix A in table form

2.2.3 Feasibility of mathematical program

Our mathematical program is designed the way to have a solution if and only if there is an arbitrage betting opportunity; otherwise, there exists no $profit \geq 0$ to satisfy all of the inequalities and there is no solution therefore, the linear program is infeasible. Otherwise, the $profit^*$ says what our profit is independent of the match outcome.

2.2.4 Max-min Classification

The solution of the Linear program defined above can be described as the maximization of minimal profit, which is the best approach for uninformed betting.

2.3 Weighted Mathematical programming model

The weighted Mathematical programming model is an extension of the previously defined Max-min.

$$
\begin{array}{ll}
\max & w^T z \\
\text{subject to} & Ax - z \geq 0 \\
& z \geq 0 \\
& x \geq minbet * y \\
& x \leq maxbet * y \\
& y \in \{0,1\}^n \\
& x \in [0,\infty)^n \\
& z \in [0,\infty))^{(points+1)^2} \\
& A \in \mathbb{R}^{(points+1)^2 \times n}
\end{array}
$$

Variable $profit$ was replaced with vector z which represents profit per match outcome, whereas $profit$ was global across all of the match outcomes. This allows us to create a new objective function that can take additional information in the form of weights.

2.3.1 Weights

The addition of weights gives us another parameter for a linear program. Weights can be chosen basically arbitrarily, however, one particular way is to set weights to be a probability distribution over possible match outcomes. In that case, the objective function behaves as an expected profit on that particular distribution.

2.3.2 Probability distributions

Many probability distributions can be used as weights. The only constraint is that distribution must be able to give a prediction for every possible match outcome. Here are two examples of distributions that we used for testing

Prior match distribution

The prior distribution is the distribution that was calculated from historical data, by counting how many times each result occurred, and then by normalizing, we get probability distribution. This distribution is very simplistic,

and it will be interesting to see whether this easy distribution can beat the max-min variant.

Independent distributions

This distribution uses two different distributions, one for home team goals and the other for away team goals. Now if we add the assumption of independence of home and away team goal counts, we can get a particular result probability $p(h, a)$ by simply multiplying the probability of the home team scoring h goals $p_h(h)$ and the same for away team and probability $p_a(a)$, which is exactly what Equation 2.4 shows.

One particular distribution, that is often used within similar situations is Poisson Distribution with probability mass function given by 2.5. In research, Poisson distribution was used in [MŠŤ14] for ice hockey predictions. The only thing needed to apply this distribution is to have lambdas for each team, ideally to have two, because of home advantage as shown for example in [KN08]

$$p(h, a) = p_h(h) * p_a(a) \tag{2.4}$$

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!} \tag{2.5}$$

2.4 Relationship between two programs

In summary, we defined two slightly different mixed integer linear programs, each defined for a different scenario. Max-min program is optimal for uninformed situations as it guarantees a good profit for each possible outcome. Whereas a weighted program is more of a greedy approach, that should work well in situations where our estimate of probabilities is more precise than the bookmaker, allowing us to take advantage of that and generate better results. However, in case of worse estimates of probabilities than those of the bookmaker, performance should be worse. The tricky part is that we can never tell which programs should be used in advance.

2.4.1 Transformation of Weighted program

Even though programs are designed differently, they still choose an optimal solution from the same space. Space of all the solutions that have a positive return on all possible outcomes. In certain cases, the weighted program can be transformed to find the same solution by setting the correct weights for the objective function. To achieve this, we need to ensure that the sum of weights is the same for all possible outcomes. A simple example is arbitrage with just a 1x2 market. If we set weights for all home-win and away-win outcomes to be one because there is the same amount of them in our goal, count representation. The weights of the drawn outcome need to be set to $\frac{|win|}{|draw|}$ and $|win| = \frac{points^2+points}{2}$ and $|draw| = points + 1$. Similar can be done in other markets and in more complex situations.

Chapter 3

Relaxations of Mathematical Programs

In the previous chapter, we defined a mathematical model for finding arbitrage opportunities, a max-min program, and a weighted model. All these work with the assumption that no matter what the outcome is profit is always positive. In this chapter, we will relax on the condition with new ideas and adjustments to previously defined models.

3.1 Motivation for relaxations

Pure arbitrage opportunities are rare; the first motivation, therefore, is to increase the amount while simultaneously not decreasing profit per bet. This can be done by enlarging the solution space via relaxing constraints. Larger solution space should also contain different optimal solutions, and by selecting the correct weights, even profit per bet could potentially improve.

3.2 Main relaxation variants

We selected two main relaxation methods for future testing. The p-arbitrage program and negative-outcome program.

3.2.1 P-arbitrage program

Given a probability distribution P over match outcomes and probability p, p-arbitrage is the relaxation that removes some rows(match outcomes e.g., 4:5) from matrix A in a way that the sum of probabilities of remaining results is greater or equal to p. Formal definition is shown in Equation 3.1.

$$\sum_{i=1}^{|outcomes|} P(r_i) \geq p \tag{3.1}$$

Removal strategy

The amount of combinations is too large for us to test all. Therefore, The central part of this relaxation is the selection of outcomes for removal. To illustrate how many combinations there are, let us assume uniform distribution over results, $p = 0.5$, and a maximum number of points 3, which means $|outcomes| = 16$. In this case, we are selecting eight arbitrary outcomes out of 16. The number of combinations is then easily determined by calculating the corresponding binomial coefficient. Twelve thousand is quite a lot alone, but if the more appropriate amount of maximum points is 10, then the binomial coefficient is around 10^{35}. Even though this illustration uses $p = 0.5$ and uniform distribution, it can still be used as an upper bound.

$$\binom{|outcomes|}{\frac{|outcomes|}{2}} = \binom{16}{8} = 12870 \tag{3.2}$$

Now that we know that trying every possible reduced matrix A is not feasible, we could search for such outcomes to reach as close to p as possible. Such a problem is well-known as the subset sum problem. The subset sum problem is defined in the following way: Given a set S of n positive integers $(s_1, s_2, \ldots, s_n)$ and a target sum T, the Subset Sum problem asks whether there exists a subset $A \subseteq S$ such that the sum of elements in A is equal to T.

Formally, we are looking for a subset $A \subseteq S$ satisfying the equation:

$$\sum_{s_i \in A} s_i = T$$

Sadly, the subset sum problem is NP-hard, as shown in [KT06]. Therefore it is not feasible for us to calculate solutions for every match independently. So we need to resort to a different approach. One typical approach used in similar cases is the greedy approach. We sort probabilities from highest to lowest. Then we add probabilities one by one until their sum does not reach or slightly overgrow p. Lastly, outcomes whose probabilities are not in the final list are removed, and a new matrix A is created. This algorithm is similar to the one used in [CKP00] for proof that it is a 1/2 approximation algorithm.

Connection to non-relaxed versions

This approach can be used with the program's max-min and weighted sum variants without any problem. The only difference is that matrix A has a different dimension. In original formulation is $A \in \mathbb{R}^{(points+1)^2 \times n}$, whereas in p-arbitrage $A \in \mathbb{R}^{|r| \times n}$, where $|r|$ is number of outcomes left after removal.

3.2.2 Negative outcome program

Another approach to relaxation is to allow negative profit for some of the match outcomes. This relaxation is only available for the weighted sum program because it would not make sense for max-min. After all, the optimal solution would either be not to bet all or to have the same negative for all match outcomes, which would not be helpful.

For the above reasons, we will describe the negative outcome program as an adjustment to the weighted sum program.

3.2.3 Adjustments

Below is the adjusted program. The only difference is that we allowed z to be negative. We also introduced a new constant threshold that will enable us to set how negative a result can be. If the *threshold* is, high possible negative profits are also high, which creates a high-risk high reward situation. A low *threshold* is a safer approach with less potential for higher yields. The optimal setting depends on the quality of probability distribution behind the

weights. Intuitively better distribution means we can allow more negative results because the expected value from the objective function is closer to the actual expected value.

$$
\begin{array}{ll}
\max & w^T z \\
\text{subject to} & Ax - z \geq 0 \\
& z \geq -threshold * maxbet \\
& x \geq minbet * y \\
& x \leq maxbet * y \\
& y \in \{0,1\}^n \\
& x \in [0,\infty)^n \\
& z \in [-threshold * maxbet, \infty))^{(points+1)^2} \\
& A \in \mathbb{R}^{(points+1)^2 \times n}
\end{array}
$$

Chapter 4

Data sources

In this chapter, we will describe all the data sources we used and how they are connected.

For our purpose, we need two connected data sets, one with as many odds as possible and another with match results, to analyze and test our programs.

4.1 Live odds

In general, one of the most critical parameters of the betting data set is time synchronization over markets. If odds from data are not synchronized well, the data does not represent the market well, and everything connected to the data is less accurate.

In arbitrage betting, synchronization is even more critical because the time window when an opportunity is opened is short.

4.2 Search for an API

The easiest way to acquire odds would be directly from the bookmaker via API(Application Programming Interface). For our needs, we would need more APIs and merge them. However, if we could do that, our whole data pipeline would be solved as bookmaker API would probably offer more than live odds.

Sadly we were able to find only a few APIs, and those were either limited or paid. This means we must go a different route for acquiring the data. Those few we found are listed in list A in the Appendix A.

4.3 Odds aggregating site

Since we could not find a suitable API, another possibility is to get odds from a odds aggregating site, which are online platforms that collect and display odds from multiple bookmakers. Everything we needed in terms of data would be in one place in structured or semi-structured form, which would be comfortable to scrape and extract from.

4.3.1 Betexplorer website

The first site we tried to get data from is [bet23]. Betexplorer is a website that compiles odds from 19 bookmakers in a semi-structured format, making it a convenient source for comparing odds. The website also offers match results, which is helpful for us in analyzing odds and results from the same source. However, the website has some limitations, including the need to open separate betting sites for placing bets and the imprecise nature of the odds. After testing the website and discovering inconsistencies in the odds, we were once again forced to switch to a different service as a source of our data.

4.4 Betting brokers

One such service is a betting broker, an online service that enables betting transactions between bettors and bookmakers. This resolves one of the problems we had with Betexplorer, not being able to place a bet in the potential future. The fact that it is available to bet on the site automatically fully fills the time synchronization requirement, as the site needs to update the odds as soon as the bookmaker changes.

4.4.1 Finding a right broker

There are many betting brokers on the internet, and for our usage, we needed a broker that had a site easy enough to crawl and was updating odds fast enough. We tried a few brokers, and in the end AsianOdds [Asi23] became the broker of our choice.

4.5 Match results

Using a betting broker has many advantages described above; however, often, brokers do not offer match results in a structured way. For any analysis, we needed results, so we had to get separate sources to match the results. However, different sources of odds and match results created a few problems that needed to be resolved.

4.5.1 Different team names

Both Betexplorer and Asianodds use different strings as team names. To use these data sets together, we need to create a mapping between these strings.

Mapping

Mapping between Betexplorer and Asianodds team name strings was more challenging than anticipated, as the first fully automated approach was implemented, which mapped teams from one data set to the second data set by some string distance metric. Many metrics were tested, including the most known Levenshtein and Hamming distances. It worked decently, but not well enough. So semi-automated approach was taken, where a few candidates were pre-selected by calculating Leveinshtein distance and manually choosing the best one by hand. This was quite tedious work, but it yielded the best results.

The main problem with a fully automated approach was that the strings representing the same team were sometimes surprisingly different. Sometimes the international name(e.g., Sparta Prague) was used. Other times local name(e.g., Sparta Praha) was used. And other inconsistencies that lead to worse performance of the automated approach. Below is the pseudocode for the Levenshtein distance.

Algorithm 1 Levenshtein Distance

```
1:  function LevenshteinDistance(str1, str2)
2:      m ← length of str1
3:      n ← length of str2
4:      Create a matrix dp of size (m + 1) × (n + 1)
5:      for i ← 0 to m do
6:          dp[i][0] ← i
7:      for j ← 0 to n do
8:          dp[0][j] ← j
9:      for i ← 1 to m do
10:         for j ← 1 to n do
11:             if str1[i] = str2[j] then
12:                 cost ← 0
13:             else
14:                 cost ← 1
15:             dp[i][j] ← min(dp[i−1][j]+1, dp[i][j−1]+1, dp[i−1][j−1]+cost)
16:     return dp[m][n]
```

4.6 Data pipeline

Now that we described what data sources were used and how to live, odds and match results are mapped together. Now we can go into more detail

about how data is gathered and stored. Two crawlers are gathering the data, one crawler for each data source.

First is the AsianOdds crawler, designed to crawl data in about three-minute intervals and save the data to a relational database. This database consists of one table with all found matches with a unique id for each match. The id is also a foreign key for searching individual odds. Every market is saved as a separate table, and among the id mentioned before, it also has a timestamp column to help join the data together. The typical task is to join together all of the odds belonging to one match. To complete the job, first of all, a matching id has to be found. Once found, odds are added market by market via a series of joins. The first joint operation extracts odds and existing timestamps. Every following join operation filters only odds that have the same timestamp.

The second is the betexplorer crawler, which has the task of collecting match results once a day. Match results are also saved into a relational database. However, this time only one table is needed.

The last part of the pipeline is a mapping between team names. How the mapping is created is already described in the previous section. Mapping is also saved as a table in a relational database.

These three parts together give us all the data we need to perform all the tests and experiments. The pipeline works well, but it has some flaws. The biggest drawback of this system is the mapping. First, it can never be complete, as a match that was not yet mapped happens all the time. Secondly, the strings representing team names change relatively often, and for every change, the mapping of that team must be remade. Lastly, the creation of mapping not being fully automated is a problem.

4.7 Sharp vs. Soft bookmakers

In the first chapter, we already described the concept of sharp and soft bookmakers. For arbitrage betting, we want as many bookmakers as possible. However, betting brokers typically offer only sharp bookmakers, which is also the case of [Asi23]. The advantage of using only sharp bookmakers is that sharp bookmakers will not limit us. However, sharp bookmakers generally have more consistency, which means that arbitrage opportunities are less likely to occur when only sharp bookmakers are considered. This fact will

influence the results toward a more negative side, but it will also better represent real-world scenarios if we apply the methods in the long term.

Chapter 5

Experiments and Results

In this chapter, we will take all the methods and approaches we defined in the previous chapters and put them to the test. Firstly in the theoretical environment to test limitations and prove the concept of relaxations. After that, we will test programs on the actual data.

5.1 Theoretical part

5.1.1 Environment for the tests

We pretend to know all the variables and distributions for this part and see how our programs behave.

More specifically, we pretend that we have three different distributions real distribution, bookmaker distribution, and our model distribution. Actual distribution simulates real life, and the results of the matches directly follow from this distribution. Bookmaker distribution is a distribution from which odds are created, and our model distribution creates weights, etc., for our programs. This experiment aims to skew distribution apart each other and see how the results are affected.

Each distribution will be Poisson used in the same way as defined before.

Using the independence of scored goals as in 2.4 and having probability mass function for home and away separately in the form of 2.5. In other words, we have two numbers for each distribution one represents how many goals the home team score on average, and the other is the same but for the away team. Skewing the distribution then means changing those two numbers.

Finally, evaluating these experiments is relatively easy because we can calculate expected values w.r.t actual distribution, which is the real expected profit.

5.1.2 The most straightforward case

In this case, we will use a 1x2 market, where everything is easier to see. This case is meant to show the value of relaxations in the case where pure arbitrage cannot be found. For all the experiments same constants were set $maxbet = 1000$,$minbet = 0$,$points = 10$, and weights are based on our model lambdas. Expected profits are calculated on more values of *threshold* to showcase the trend.

5.1.3 All distributions are the same

Nothing is interesting when all distributions are the same; the optimal solution for all programs is not to bet all. More precisely is that it is not enough if the bookmaker and our model use the same distribution; in that case, the bookmaker has the edge in the form of the margin, and there is no value to be gained.

5.1.4 Bookmaker distribution is more accurate

First, when we say more accurate, we mean closer to actual distribution. We do that in our testing environment by setting lambdas in the following way. We set both home and away lambdas for a bookmaker to the real lambdas, and we skew lambdas of our model by some delta. Unsurprisingly when bookmaker odds are more accurate, the real expected profit is negative. Figure 5.1 shows how expected profit changes when *threshold* grows.

5.1.5 Our model is more accurate

Here we swap the lambdas of the bookmaker and our model from the previous example, and the expected profit is positive. Figure 5.2 shows once again the expected profit in dependency with *threshold* value.

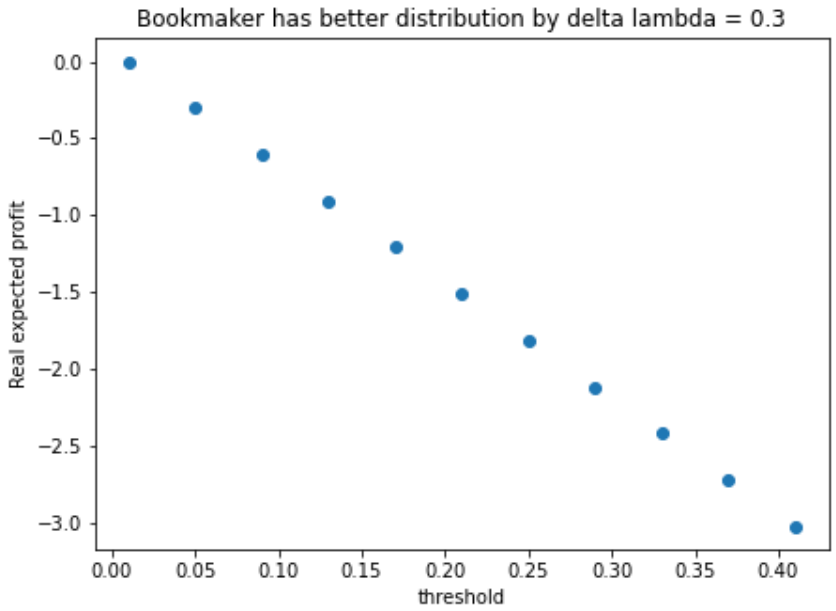


Figure 5.1: Showing loss when a bookmaker has better odds

5.1.6 Experiment summary

In summary, this experiment shows that programs behave the way we imagined.

5.2 Double Poisson distribution

In this section, we will go into more detail about the model that we use to calculate weights for our weighted model and the probabilities we need in the negative outcome program or in the P-arbitrage program. We have already suggested in Figure 2.3.2 usage of two independent Poisson distributions. This distribution is often called double Poisson distribution. Calculating the

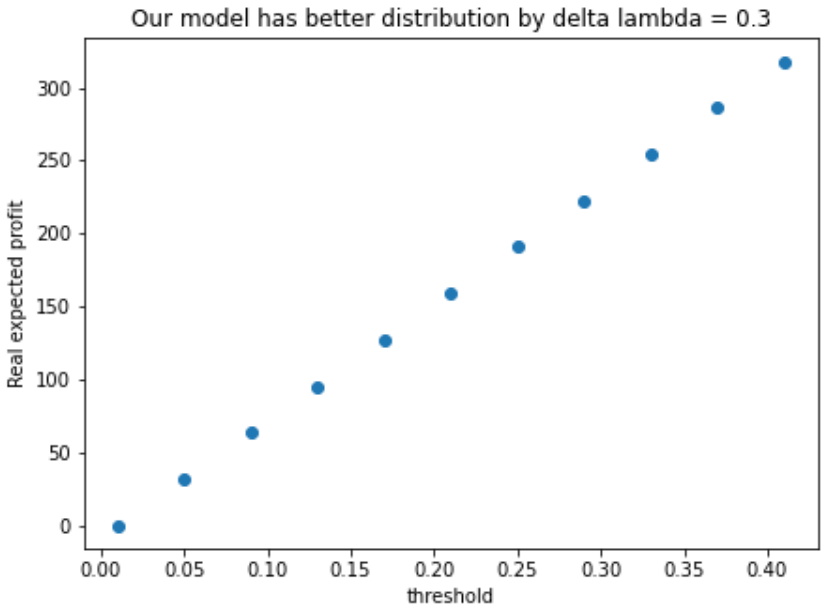


Figure 5.2: Showing pure profit when the model has better odds

exact result is easy enough, given both lambdas. For calculating different probabilities, the Skellam distribution can be used. Skellam distribution models the difference between two random Poisson distributed variables, which can be used to determine the probability of a home team winning.

5.2.1 Accuracy of model

Authors in [H21] found the accuracy of the double Poisson model to be 48 %. Just for clarification, accuracy in this context means the ratio of football matches it predicted(by choosing the most probable outcome) correctly.

After fitting the model on our data, we achieved an accuracy of 42 %, which is quite worse than the authors mentioned above. This means that it brings some additional information to the program. However, if it is enough will be seen after experiments are run.

5.3 Real data tests

Now we move on to the real data test. In real data tests, we must define different metrics for result comparison. A prevalent metric in this domain is the return on investment(ROI), a ratio of profit and the total amount wagered.

$$ROI = \frac{profit}{\sum_{i=1}^{n} x_i} \tag{5.1}$$

The tests were performed on data set with about 40000 matches out of a possible 50000. The 10000 could not be used because results were missing or mapping was not done for the teams.

5.4 Pure arbitrage opportunities

5.4.1 Existence of pure arbitrage opportunities

One of the motivations for creating relaxation was to enlarge the number of opportunities to bet on. Hence we will start with an analysis of how many arbitrage opportunities and how long each opportunity lasts. This exactly is shown in Figure 5.3. The majority of opportunities last only up to 3 minutes(that is our refresh interval). Few last 3-6 minutes, and more extended opportunities are relatively rare.

The figure also shows that we were able to find around 100 matches that contained a betting opportunity. That is approximately 1 in 400 games, which is about as expected.

5.4.2 Results

Before we can move to test results, we need to talk about evaluation and testing strategy, metrics, and setting of parameters.

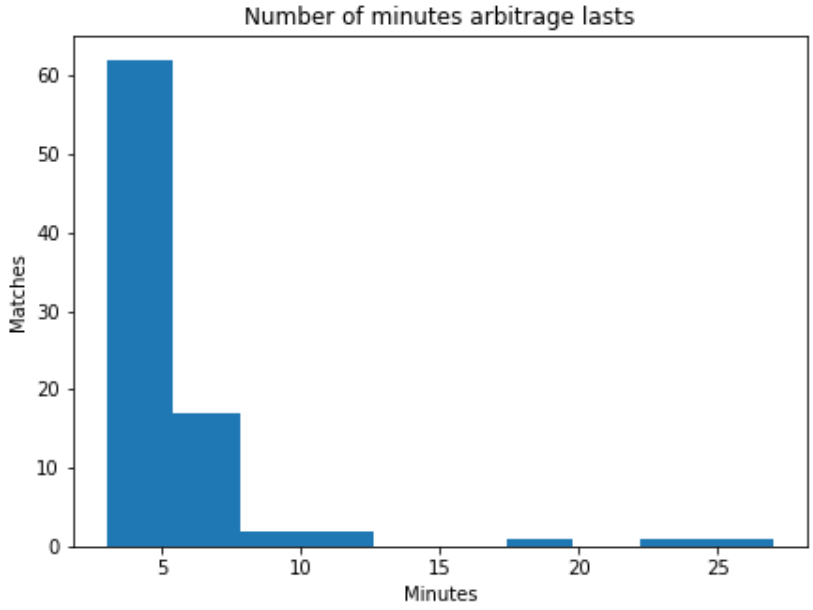


Figure 5.3: Shows how long an arbitrage lasts

Parameters are set at $maxbet = 1000$, $minbet = 0$, $points = 10$, and weights are set by prior distribution or lambda distribution.

Metrics are easier in the case of pure arbitrage opportunities because we do not need to take into account a number of opportunities, as they are all the same. So, in this case, we can use pure profit as a metric for comparison. Return on investment or similar economic metrics can be used as well. However, the results would be all the same. For some deeper analysis, it also helps to consider the individual distributions of profits.

Now we can finally move to results. Figure 5.4 shows a box plot that compares the distribution of profits per individual bet. When we compare the individual distribution, it corresponds well with expected behavior; the basic variant is more reliable. It has lower variance, whereas lambda and prior variants' greedy search for expected value often end up in lower outcomes. The idea behind lambda and prior variants are to win once in a while big to raise that expected value. Using these values, pure profit is 38822 for the lambda variant, 32859 for the prior variant, and 29368 for the basic variant, so in this case, lambda seems like the best approach.

However, the advantage is purely given by a few high-profit outcomes that could also be treated as outliers. If we remove the most obvious outliers, we get Figure 5.5 if we remove the most obvious outliers. In this case, the basic

variant is clearly the best. Pure profits are 17500 for the lambda variant, 17000 for the prior variant, and 29368(same as before) for the basic variant. This shows the biggest weakness of maximizing the expected value approach, not considering variance.

In the weighted sum program, we can modify constraint $z \geq 1$ to a larger number on the right hand to lower the variance. This variant was also tested, but it did not yield better results. The same goes for the minimum bet constant can also be modified for possible lower variance.

this section, we tried two appThe comparison was relatively straightforward for calculating classical risk-free bets. While the weighted model showed the best performance on our data, some of the points showed signs of typical outliers. After removing outliers, basic methods showed the best results.

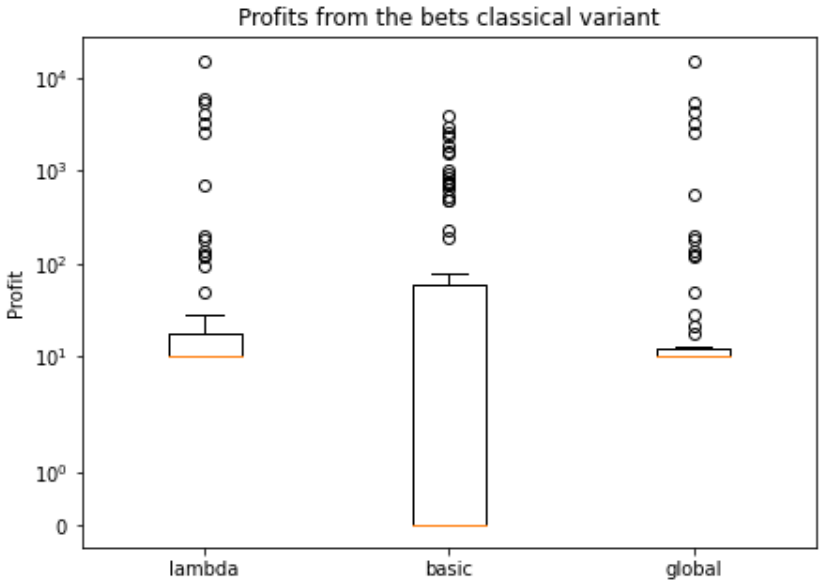


Figure 5.4: Box plot of pure profits with outliers

5.5 Relaxed versions

In the previous section, we tested programs calculating classical risk-free bets. Since all of the programs calculate the same thing, just with different objective functions. Compared to relaxed versions, the comparison is not easy, as relaxation can also be combined, and their approach is more different.

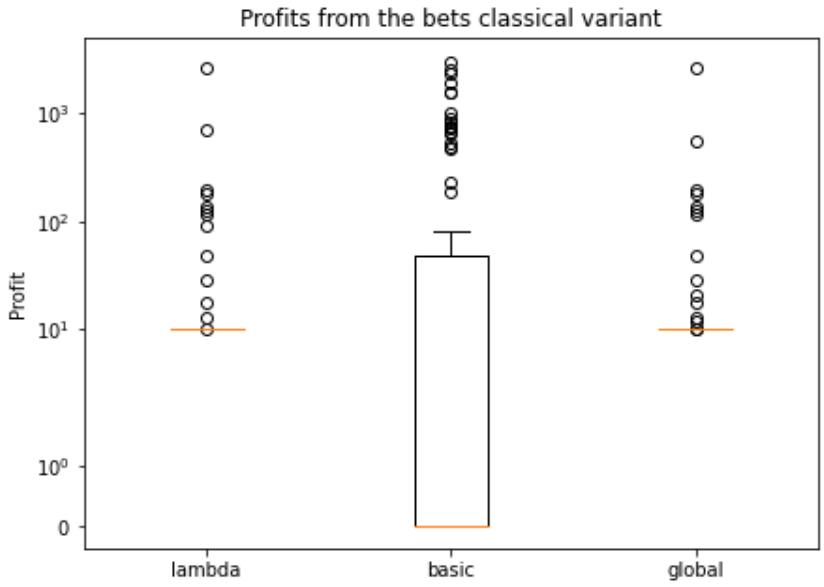


Figure 5.5: Box plot of pure profits without outliers

Also, an important part of relaxation is not just how profitable it is but the number of opportunities in comparison to pure arbitrage number. Also, all results are presented after outlier removal unless stated otherwise.

5.5.1 Negative outcome program

The negative outcome value program relaxes the part of the weighted sum program by allowing negative profits for match outcomes by setting the *threshold* parameter to a positive number. This allows parameter z to be negative, which also means that every set of odds will have a trivial solution of betting vector $x = 0$. Programs that had such a solution will be ignored in the result analysis.

Threshold range

We decided to test the program on values of *threshold* from 0.01 to 0.2. The value of the threshold gives us control of the risk taken. Variable *threhold* can be directly interpreted as a maximum percentage of max bet that can be lost.

Number of opportunities

One of the motivations for relaxation was to increase the number of opportunities significantly. The number of opportunities was constant at 1365 for every setting of the threshold. This seems to show that increasing the threshold only changes the betting vector or strategy. The number of opportunities is significantly higher than in the case of the pure arbitrage part. In terms of the number of opportunities, relaxation helped a lot, more specifically, from around 100 to 1365.

Profits

The distribution of profits can be seen in Figures 5.6 and 5.7. The box plot shows well how with a higher threshold value, more negative outcomes start to appear, and the same applies to higher-value bets. This is expected behavior and confirms that model at least works as intended.

The Sum of profits is shown in Figure 5.8 and shows that our model does not work well. The higher the threshold higher the loss is. Regarding return on investment, there are different thresholds close to each other as higher risk allows higher total bets. ROI is, of course, also negative but a little smaller than the bookmaker margin, which means that at least some additional information was added by using this approach.

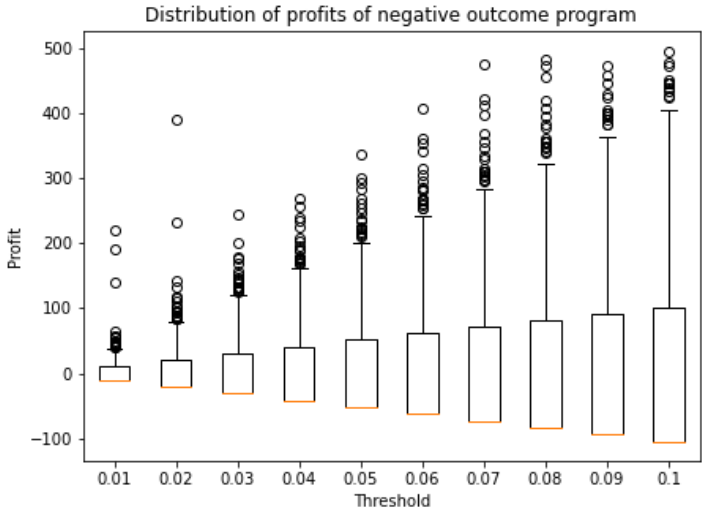


Figure 5.6: Box plot of pure profits of the first half of negative outcome program

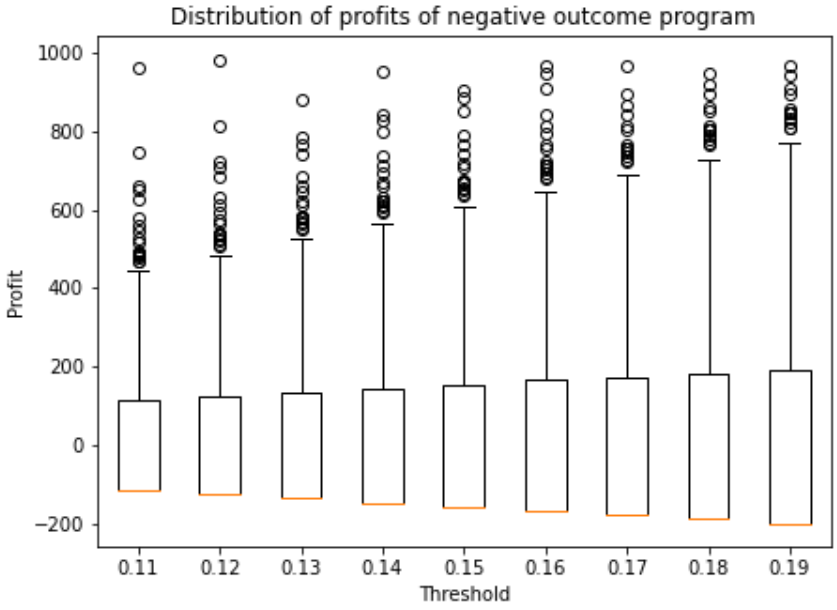


Figure 5.7: Box plot of pure profits of the second half of negative outcome program

Prior distribution

Tests from previous sections were done on double Poisson distribution, we also did a test negative outcome program using prior distribution, and the results are similar but a bit worse.

Summary

The negative outcome program did well regarding the number of opportunities. However, additional opportunities yield negative profit. This means this method is not well suited for practical usage.

5.5.2 P-arbitrage program

This approach focuses on removing some rows from matrix A, making the problem less constrained in the hope of gaining additional opportunities

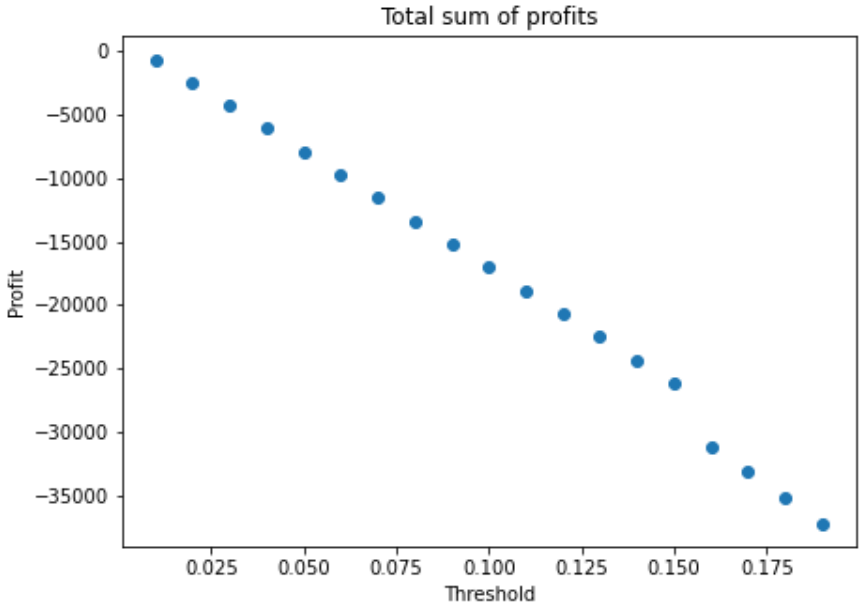


Figure 5.8: Sum of profits per threshold

without losing profit. As stated in one of the previous chapters, the removal is done by a greedy approach. Also, after the removal, matrix A is put in the max-min program in the test below.

Probabilities range

For the tests, we chose probabilities p of 0.75,0.8,0.85,0.9,0.95. Higher probabilities should be a more safe approach but should also generate a smaller number of opportunities.

Number of opportunities

Several created opportunities can be seen in Figure 5.9. The increase is less significant than in the negative outcome approach. This is intuitive because removing a few rows from matrix A does not increase possible space much.

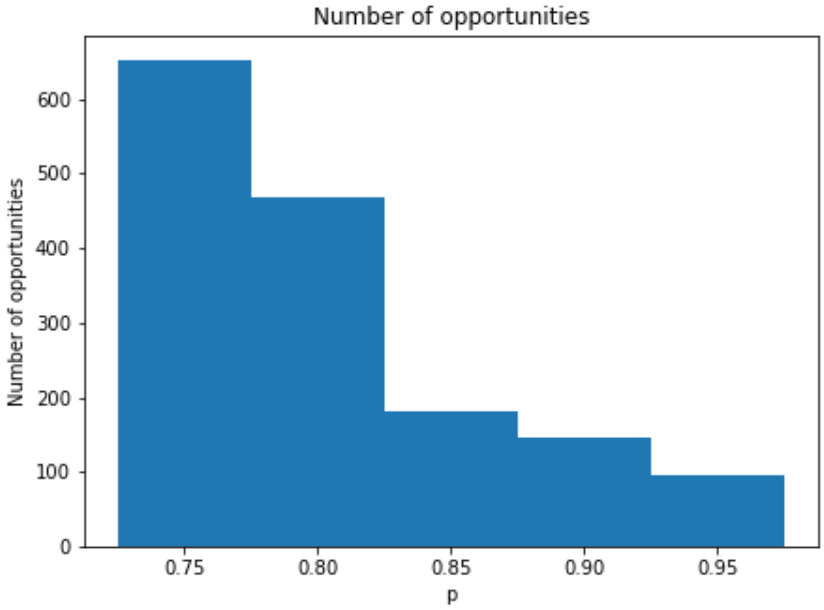


Figure 5.9: number of opportunities for each threshold

Profits

In this case, profits are even more sensitive to outlier detection, as shown in Figure 5.11. For clarification, a figure with strict outlier detection does not show $p = 0.75$ because it was too negative. From profits alone, it can already be seen that results are much more positive than in the case of a negative outcome program. The distribution of profits is in Figure 5.10 with already removed outliers. Overall this approach performs better in the profit part, but the increase in the number of opportunities is not that significant.

5.5.3 Real-world arbitrage betting system

For additional data for comparison, we tried one of the real-world arbitrage betting systems RebelBetting, with website [Reb23]. All the results are from the one month we used the service.

One advantage of RebelBetting was that it offered AsianOdds as one of the available sources of odds. This allows comparison with the number of opportunities for each approach. We also selected additional soft bookmakers to see how several opportunities change with the addition of a soft bookmaker.

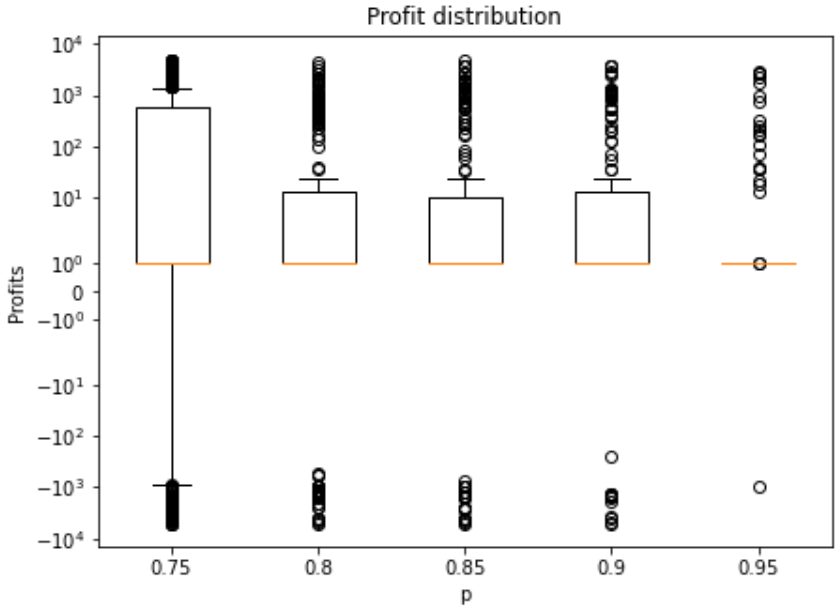


Figure 5.10: Profits for each threshold

To our surprise, RebelBetting did not find any pure arbitrage opportunities with AsianOdds alone; all opportunities found were combinations of AsindOdds and the soft bookmaker. Figure 5.12 shows profits from around 80 found opportunities. A direct comparison to our model cannot be made because the system does not use the max bet system as we do; instead, it has a predetermined bankroll divided into bets.

We can see that all the bets have only positive returns, which means that the system probably does not use any relaxation. Additionally, this test shows how helping is to have at least one soft bookmaker's odds available. Lastly, RebelBetting covers the three largest bet markets Asian Handicap, 1x2 market, and Over-Under.

Also, every found opportunity was one of the three possible scenarios. The first scenario is to bet on all of the 1x2 market outcomes. Another possibility is to bet on a home or away win and rest by an +0.5 or +0.25 Asian Handicap. An example is shown in table 5.1. It is interesting here that in the case of +0.25 Asian Handicap, it is a more complex arbitrage with overlapping markets. And final scenario is to bet on either a home win or an away win plus a draw, and the rest is covered by Asian Handicap + 0. An example is in Table 5.2. There is also one bet on under and over 2.5. None of the bets covered more than three markets, which is one of the differences to our approach when multiple bets used more than three markets. However,

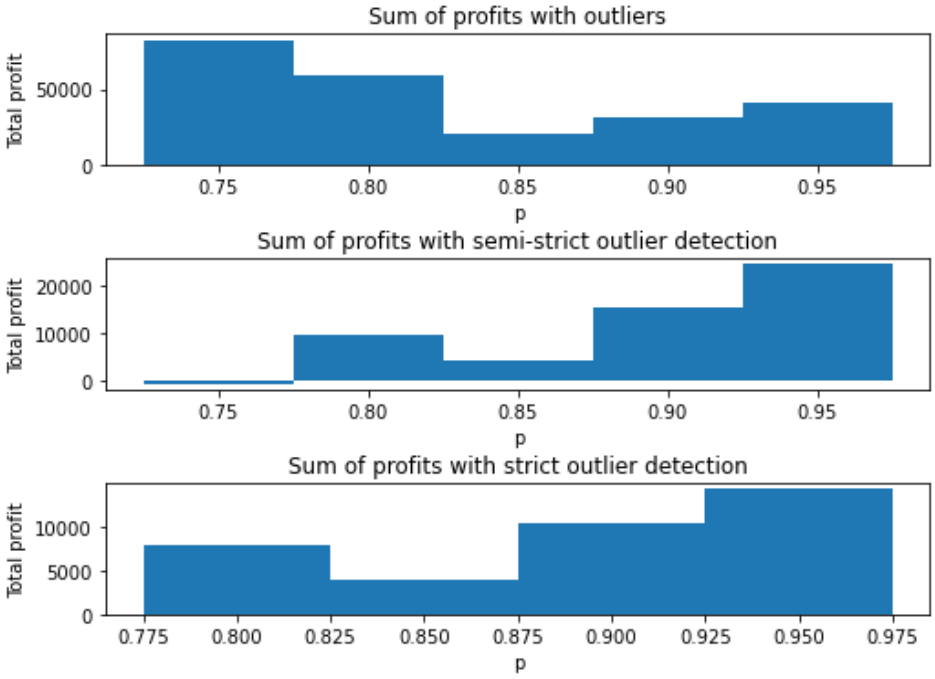


Figure 5.11: Sum of profits for each threshold with three levels of outlier detection

those bets did not lead to any improvement. More often than not, those bets were losing ones.

RebelBetting betting strategy seems similar, if not the same, as our max-min program. When we run our max-min program on the odds found by RebelBetting. The only difference is that RebelBetting is, by default rounding bets to tens (possibly to simulate more human behavior and avoid limitation), whereas we round to just whole numbers.

In summary, we examined one of the most used arbitrage betting systems called RebelBetting. The system searched for arbitrage opportunities on AsianOdds, the same site as we used, plus one additional soft bookmaker. The system did not find any pure AsianOdds opportunities. Most of the found opportunities were simple ones without any market overlap, but the system also showed that it could search for more complex ones with Asian Handicap +0.25 example. The betting strategy seemed to be close to our max-min program.

		Home Goals				
		0	1	2	3	4
Away Goals	0	Ah +0.5	Ah +0.5	Ah +0.5	Ah +0.5	Ah +0.5
	1	Away	Ah +0.5	Ah +0.5	Ah +0.5	Ah +0.5
	2	Away	Away	Ah +0.5	Ah +0.5	Ah +0.5
	3	Away	Away	Away	Ah +0.5	Ah +0.5
	4	Away	Away	Away	Away	Ah +0.5

Table 5.1: Scenario 2

		Home Goals				
		0	1	2	3	4
Away Goals	0	Draw	Ah +0	Ah +0	Ah +0	Ah +0
	1	Away	Draw	Ah +0	Ah +0	Ah +0
	2	Away	Away	Draw	Ah +0	Ah +0
	3	Away	Away	Away	Draw	Ah +0
	4	Away	Away	Away	Away	Draw

Table 5.2: Scenario 3

5.6 Summary of test results

The classical max-min program was the best-performing(generated the most profit) method. Weighted-sum program did not lead to any improvement. For the relaxations, the negative outcome program significantly enlarged the number of opportunities, but profit was negative, so a larger number of opportunities is useless. The other relaxation p-arbitrage program performed better in terms of profit and increased the number of opportunities. Still, the average yield per bet decreased to the point that it generated less profit in total than the max-min program.

5.7 Outliers

During testing, we encountered two types of outliers. First are high-profit(more than 5000) outliers, which are caused by, an unlikely event occurring. One example is the winning of the team with odds of 33. The odds being this high is really unusual, which is one possible indicator of it being an outlier. Another possibility is a mistake with timestamps and us interpreting live odds as non-live. Live odds have a different meaning and cannot be interpreted directly independently of the live score. The other type of outlier is very

Figure 5.12: RebelBetting profits

negative profits. These mainly occur in p-arbitrage relaxations with lower probability p, in this case, it is even more likely that they occurred because of misinterpretation of live odds.

Because we are unsure if these data points are real, we mostly report both with outliers and without outliers results. But in the end, the results without outliers are more realistic.

Chapter 6

Conclusion

In conclusion, this thesis has explored the application of mathematical programming in the context of arbitrage betting. However, the results obtained from this research have been underwhelming, indicating limitations and challenges associated with using mathematical programming techniques in this domain.

Despite the initial expectations and the potential perceived in mathematical programming, this thesis's empirical analysis and findings suggest that identifying profitable arbitrage opportunities solely through mathematical models may be more difficult than anticipated. The sports betting market has become increasingly efficient, with bookmakers adjusting their odds quickly to minimize potential arbitrage opportunities. As a result, the gaps in odds necessary for successful arbitrage betting have become scarce and short-lived.

The underwhelming results of this thesis highlight the need for a more comprehensive approach to arbitrage betting, one that incorporates not only mathematical programming but also qualitative analysis and consideration of market trends. While mathematical programming alone may not yield the desired outcomes, it still holds potential when used in conjunction with other analytical methods. Integrating machine learning algorithms, data mining techniques, and advanced statistical models may enhance the effectiveness of arbitrage betting strategies by providing better weights for our programs.

6.1 Future work

Although we tested multiple different strategies in this thesis, more possible approaches could be leveraged to create more positive outcomes.

One possible expansion of our model is to apply modern portfolio theory to better handle variance, especially in the weighted program, that greedily goes for one outcome.

Another possibility is to try using a more accurate model for the weight calculation. Our double Poisson model has only 42 % accuracy, and several works showed higher accuracy on the models.

Expansion of available bookmaker sets is always possible, as RebelBetting showed that having more bookmakers means more opportunities, and the max-min program would alone be enough.

Bibliography

[Aba89] Jeph Abara, *Applying integer linear programming to the fleet assignment problem*, Interfaces **19** (1989), no. 4, 20–28.

[Asi23] AsianOdds, *Asianodds*, `https://www.ao0188.com/Login.aspx`, accessed 2023.

[bet23] betexplorer, *betexplorer*, `https://www.betexplorer.com/`, accessed 2023.

[BFGM00] Huguette Beaulieu, Jacques A Ferland, Bernard Gendron, and Philippe Michelon, *A mathematical programming approach for scheduling physicians in the emergency room*, Health care management science **3** (2000), 193–200.

[Buc03] Joseph Buchdahl, *Fixed odds sports betting: Statistical forecasting and risk management*, Summersdale Publishers LTD-ROW, 2003.

[CDLR02] Martin Crowder, Mark Dixon, Anthony Ledford, and Mike Robinson, *Dynamic modelling and prediction of english football league matches for betting*, Journal of the Royal Statistical Society: Series D (The Statistician) **51** (2002), no. 2, 157–168.

[Chv83] Vasek Chvatal, *Linear programming*, Macmillan, 1983.

[CKP00] Alberto Caprara, Hans Kellerer, and Ulrich Pferschy, *The multiple subset sum problem*, SIAM Journal on Optimization **11** (2000), no. 2, 308–319.

[CKS90] William Cook, Ravindran Kannan, and Alexander Schrijver, *Chvátal closures for mixed integer programming problems*, Mathematical Programming **47** (1990), no. 1-3, 155–174.

[CLP03] Michael Cain, David Law, and David Peel, *The favourite-longshot bias, bookmaker margins and insider trading in a variety of betting markets*, Bulletin of Economic Research **55** (2003), no. 3, 263–273.

[Dan02] George B Dantzig, *Linear programming*, Operations research **50** (2002), no. 1, 42–47.

[DF8] Christian Deutscher, Bernd Frick, and Marius Ötting, *Betting market inefficiencies are short-lived in german professional football*, Applied Economics **50** (2018), no. 30, 3240–3246.

[FNV09] Egon Franck, Stephan Nuesch, and Erwin Verbeek, *Inter-market arbitrage in sports betting*, Tech. report, National Centre for Econometric Research, 2009.

[FVN13] Egon Franck, Erwin Verbeek, and Stephan Nüesch, *Inter-market arbitrage in betting*, Economica **80** (2013), no. 318, 300–325.

[GT91] Joseph Golec and Maurry Tamarkin, *The degree of inefficiency in the football betting market: Statistical tests*, Journal of Financial Economics **30** (1991), no. 2, 311–323.

[HA10] Lars Magnus Hvattum and Halvard Arntzen, *Using elo ratings for match result prediction in association football*, International Journal of Forecasting **26** (2010), no. 3, 460–470, Sports Forecasting.

[HW23a] Tadgh Hegarty and Karl Whelan, *Disagreement and market structure in betting markets: Theory and evidence from european soccer*.

[HW23b] ———, *Forecasting soccer matches with betting odds: A tale of two markets*, Tech. report, CEPR Discussion Papers, 2023.

[HZ90] Donald B Hausch and William T Ziemba, *Arbitrage strategies for cross-track betting on major horse races*, Journal of Business (1990), 61–78.

[H19] Ondřej Hubáček, Gustav Šourek, and Filip Železný, *Exploiting sports-betting market using machine learning*, International Journal of Forecasting **35** (2019), no. 2, 783–796.

[H21] Ondřej Hubáček, Gustav Šír, and Filip Železný, *Forty years of score-based soccer match outcome prediction: an experimental review*, IMA Journal of Management Mathematics **33** (2021).

[KN08] Dimitris Karlis and Ioannis Ntzoufras, *Bayesian modelling of football outcomes: using the Skellam's distribution for the goal difference*, IMA Journal of Management Mathematics **20** (2008), no. 2, 133–145.

[KT06] Jon Kleinberg and Eva Tardos, *Algorithm design*, Pearson Education India, 2006.

[MGOF21] Uhrín Matej, Šourek Gustav, Hubáček Ondřej, and Železný Filip, *Optimal sports betting strategies in practice: an experimental review*, IMA Journal of Management Mathematics **32** (2021), no. 4, 465–489.

[MŠŤ14] Patrice Marek, Blanka Šedivá, and Tomáš Ťoupal, *Modeling and prediction of ice hockey match results*, Journal of quantitative analysis in sports **10** (2014), no. 3, 357–365.

[Reb23] RebelBetting, *Rebelbetting*, `https://www.rebelbetting.com/`, accessed 2023.

[VDM09] Nikolaos Vlastakis, George Dotsis, and Raphael N. Markellos, *How efficient is the european football betting market? evidence from arbitrage and trading strategies*, Journal of Forecasting **28** (2009), no. 5, 426–444.

[WW94] LINDA M. WOODLAND and BILL M. WOODLAND, *Market efficiency and the favorite-longshot bias: The baseball betting market*, The Journal of Finance **49** (1994), no. 1, 269–279.

Appendix A

Available bookmaker's API

- https://pinnacleapi.github.io/
- https://www.cloudbet.com/api/
- https://betting-api.com/sbobet/