NTNU – Trondheim
Norwegian University of
Science and Technology

The Betting Machine

Using in-depth match statistics to compute
future probabilities of football match outcomes
using the Gibbs sampler

Martin Belgau Ellefsrød

Master of Science in Computer Science
Submission date: June 2013
Supervisor: Helge Langseth, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

Football is one of the most, if not *the* most, popular sporting games in the world, both played and watched by millions of people from all over the world almost daily, certainly weekly. Though most of those who place weekly bets on match outcomes have made up their minds on the abilities on competing teams, many have nevertheless attempted to assess the abilities of sporting teams using different statistical approaches, assigning objective, quantitative values to each team. From that standing point, one can then try to predict the future results of games. This paper researches the existing methods used by Maher (1982) and Dixon & Coles (1997) on modeling team strengths, and how these models are used for prediction.

The study then proceeds to compare the two methods of Maher (1982) and Dixon & Coles (1997) by experimenting with the models, finding that the latter seems to provide the most promising results. Tests are run by constructing the models and collecting empirical evidence on the accuracy on the models when using them to bet on matches.

We then continue with constructing our own model, which utilizes more detailed data from the current season's football matches, retrieved from several football and betting sites on the internet, and compare our results with how the older models performed on the same season.

Our study finds that the current data we were able to retrieve does not significantly increase the return of investments when betting on matches over the course of a season. Though our model performs slightly better than the two methods of Maher(1982) and Dixon & Coles(1997), it is not able to perform better than the bookmakers it is betting against.

The study is concluded by a section on what further work should be done to attempt to improve the models, focusing on using extensive data on matches that we did not manage to find, such as where on the pitch most passes were made, or where shots where fired from, and whether important players were available.

Preface

This project was done by a master student at the Norwegian University of Science and Technology. In the latter stages of my studies, I have selected Game Technology as my specialization, but have also completed several courses required for Artificial intelligence students, as I also have an interest in this field. I have always had a deep interest in football statistics and the appliance of this to betting strategies. As a consequence, I was drawn to this project, as it combines two fields I find very interesting. The project offered freedom in regards to how to attack the problem, but as my supervisor Helge Langseth had already constructed a framework for working with Gibbs sampling in Matlab, I chose to continue development of this framework, using this method.

Acknowledgments

I would like to thank my supervisor Helge Langseth for his immense expertise in the fields of football betting, statistical analysis and AI methods; without his help over the course of this project, it would not be what it is today. I would also like to thank my fiancée, Nikoline, who has been so very supportive these past 7 years.

Contents

1 Introduction 1

1.1 Background and Motivation . . . 2

1.2 Research Questions . . . 4

1.3 Research Method . . . 4

2 Background Theory and Motivation 7

2.1 Background Theory . . . 7

2.1.1 Markov Chains . . . 7

2.1.2 Gibbs Sampling . . . 8

2.2 Assessment of Maher, Dixon & Coles . . . 9

2.2.1 Using the Poisson Distribution . . . 10

2.2.2 Introducing the bivariate Poisson model . . . 10

2.2.3 The Home Ground Advantage and Varying Team Form . . . 11

2.2.4 Altering the Poisson Distribution by Inflation . . . 12

2.2.5 Comparision of Models . . . 12

2.3 Technologies and Workflow . . . 13

2.3.1 MATLAB . . . 13

2.3.2 JAGS . . . 13

2.3.3 HTML . . . 14

2.3.4 PHP . . . 14

2.3.5 WampServer . . . 14

2.3.6 Regular Expressions . . . 15

2.3.7 Workflow . . . 15

3 System Architecture 17

3.1 The internet Crawler . . . 17

3.1.1 Important Data Files . . . 17

3.1.2 Important PHP-scripts . . . 20

3.1.3 Taking it step by step . . . 21

3.2 The Betting Simulator . . . 26

3.2.1 Game . . . 27

3.2.2 GameList . . . 27

3.2.3 Database . . . 27

3.2.4 Simulator . . . 30

3.2.5 Bookie . . . 30

3.2.6 Footy . . . 30

3.2.7 readData . . . 30

4 Comparing the Models **33**
4.1 Maher . . . 33
4.2 Dixon and Coles . . . 34
4.3 Experiments and Results . . . 35
4.3.1 Experimental Plan . . . 35
4.3.2 Betting Strategy . . . 35
4.3.3 Experimental Setup . . . 37
4.3.4 Experimental Results . . . 40
4.4 Evaluation . . . 46

5 Constructing a Model **49**
5.1 The statistics available . . . 49
5.2 Using the coefficient of determination . . . 50
5.2.1 The predictive nature of goals . . . 50
5.2.2 Using other variables . . . 51
5.2.3 Choosing variables . . . 53
5.3 The Model . . . 54

6 Experiments and Results **59**
6.1 Experimental Plan . . . 59
6.1.1 Fixed Bets . . . 59
6.1.2 Fixed Return . . . 59
6.1.3 Only Favourites . . . 60
6.2 Experimental Setup . . . 60
6.3 Experimental Results . . . 61

7 Evaluation and Conclusion **69**
7.1 Evaluation . . . 69
7.2 Discussion . . . 69
7.2.1 Useful in-depth data . . . 70
7.2.2 Improvement of model . . . 70
7.3 Contributions . . . 70
7.4 Continuation of Work . . . 70

A **75**
A.1 League Positions 2011/2012 . . . 75
A.2 Maher model implementation . . . 77
A.3 Dixon and Coles model implementation . . . 78
A.4 Our model implementation . . . 81

B **85**
B.1 Structured Literature Review . . . 85
B.1.1 Rationale . . . 85
B.1.2 Research Questions . . . 85
B.1.3 Review Protocol . . . 86

B.1.4 Key Terms and Strings 86
B.1.5 Sources 88
B.1.6 Selected Primary Studies 89
B.1.7 Quality Assessment 92
B.2 State of the Art Assessment 93
B.2.1 Using the Poisson Distribution 93
B.2.2 Introducing the bivariate Poisson model 94
B.2.3 The Home Ground Advantage and Varying Team Form 95
B.2.4 Altering the Poisson Distribution by Inflation 96
B.2.5 Being Superior 97
B.2.6 Further Research on Team Characteristics 97
B.2.7 Other Models 98
B.2.8 Using Inadequate Scoring Rules 99
B.2.9 Comparision of Models 100

C Code Documentation 103
C.1 Data files 103
C.1.1 rawTable.txt 103
C.1.2 legacy-matches.csv 104
C.1.3 fixture-info.csv 104
C.1.4 upcoming-fixtures.csv 105
C.1.5 odds.csv 105
C.2 Internet crawler 105
C.2.1 whoscored_league_table_and_match_list.php 105
C.2.2 whoscored_match_ifo.php 106
C.2.3 get_upcoming_fixtures.php 106
C.2.4 get_odds.php 107
C.2.5 simple_html_dom.php 107
C.3 Betting Simulator 108
C.3.1 Game 108
C.3.2 GameList 108
C.3.3 Database 108
C.3.4 Bookie 109
C.3.5 Simulator 109
C.4 Collecting data from a league 110
C.5 Using the Simple_html_dom.php Script 111
C.6 Setting up the WampServer 111

List of Figures

2.1 Conditional independence of the Markov chain 7

3.1 Overview of crawler system 18
3.2 League table on www.whoscored.com 22
3.3 average odds on betexplorer.com 25
3.4 Odds for single match on betExplorer.com 26
3.5 MATLAB class diagram 28

4.1 The Maher (1982) model constructed as a Baysian network 34
4.2 Autocorrelation for Arsenal using the Maher model with thinning at 1. 38
4.3 Autocorrelation for Arsenal using the Maher model with thinning at 100. . . . 39
4.4 Autocorrelation for Arsenal using the Maher model with thinning at 30. . . . 40
4.5 A depiction of how the training set S increases with time. 41
4.6 Attacking ability of Wigan Athletic 45
4.7 Defending ability of Wigan Athletic 46
4.8 Defending ability of Manchester City 48
4.9 Attacking ability of Manchester City 48

5.1 Markov chain generated by jags with the extended model 57

6.1 Comparision of models using variance-adjusted betting strategy 63
6.2 Comparision of models using variance-adjusted betting strategy and selecting only favourites 64
6.3 Comparision of models using fixed bet betting strategy 65
6.4 Comparision of models using fixed bet betting strategy and selecting only favourites 66
6.5 Comparision of models using fixed return betting strategy 67
6.6 Comparision of models using fixed return betting strategy and selecting only favourites 68

A.1 Appendix: League positions over the course of the season, part 1 75
A.2 Appendix: League positions over the course of the season, part 2 76

List of Tables

1.1 Betting distribution and odds presented by a bookmaker for a given football game . . . 2
1.2 Expected return for a bookmaker for a arbitrary football match, given the betting distribution described in column 2. . . . 3
1.3 Probabilies presented by a bookmaker and by our model for a given football game . . . 3

2.1 List of some of the prominent metacharacters of regular expressions . . . 15

3.1 Game properties . . . 27
3.2 Database properties . . . 29

4.1 Profits for match outcomes for an arbitrary game . . . 36
4.2 Results of betting with the model proposed by Maher (1982), rounds 20-28 . . 42
4.3 Results of betting with the model proposed by Maher (1982), rounds 29-38 . . 42
4.4 Shows how the Maher(1982)-model fared when trying to anticipate the last Wigan matches of the season. . . . 43
4.5 Results of betting with the model proposed by Dixon & Coles (1997), rounds 20-28 . . . 44
4.6 Results of betting with the model proposed by Dixon & Coles (1997), rounds 29-38 . . . 44
4.7 Shows how the Dixon & Coles(1997)-model fared when trying to anticipate the last Wigan matches of the season. . . . 47

5.1 Statistical variables obtained with internet crawler. Only home-team statistcs are shown. . . . 49
5.2 R^2 values for each team for goals scored in round t opposed to round t+1 for t={1,...37} . . . 51
5.3 R^2 values for several variables in round t opposed to goals in round t+1 . . . 52
5.4 R^2 values for several variables in round t opposed to goals in *that round* . . . 52
5.5 R^2 values for several variables in round t opposed to *goal difference* in that round . . . 53
5.6 Average intensity of each team in the EPL over a season . . . 54
5.7 Average dominance of each team in the EPL over a season . . . 55
5.8 Average amount of shots on target of each team in the EPL over a season . . 56

6.1 Results of models when using different betting strategies for rounds 2 to 38 accumulated. . . . 61

6.2 The average return for each model over the six betting strategies used. 62
6.3 Predicted League Table 62

B.1 List of inclusion criteria 87
B.2 list term groups used for searching sources 87
B.3 Quality assessment statements 92

1 Introduction

This chapter firstly provides background information and motivation for the work presented in the later chapters. We will then proceed with a description of the research method used and the structure of the thesis presented.

Chapter 2 presents the work done by other authors, and examines methods and data used in these works. This is a summary of the most important literature found in the Structured Literature Review written for the preliminary project of this master thesis. This chapter also explains the Markov Chain Monte Carlo algorithm, through Gibbs sampling, which will be used for experimentation when attempting to assess the strengths of some of the approaches presented in chapter 2, as well as our own model. This background theory is also taken from the preliminary project.

Chapter 3 presents the architecture of the system we are using to obtain match-data as well as the framework for testing several predictive models. More in-depth descriptions of the system architecture can be found in Appendix C.

Chapter 4 gives a more detailed presentation of the models proposed by Maher (1982) and Dixon & Coles (1997), and presents the experimental results of comparing the two models' predictive accuracy by testing them on last years (2011/2012) English Premier League season, and comparing their return on investments. This chapter gives the main results of our preliminary project, and has been directly extracted from that report. Only slight structural adjustments have been made.

Chapter 5 describes our adapted Dixon & Coles (1997) model, using in-depth match data in addition to goals scored.

Chapter 6 gives the experiments done and results found during this project. These experiments will focus on testing our model and comparing it with two of the most promising methods presented in chapter 2 and 4, and assessing the strengths and weaknesses of each.

Chapter 7 presents conclusions and evaluations of the project results and methods, and the continuation of this work and how improvements can be made to the model to improve its predictive ability.

1.1 Background and Motivation

Football (the European version, not the American) is by many regarded as the most popular sport in the world, especially with regards to the amount of games tv-broadcasted, and attendances at local stadiums. There exists many bookmakers which accept bets on virtually any and all games at agreed upon odds. In order for such bookmakers to thrive and profit, it is essential that they are very good at setting odds for and predicting the outcomes.

A bookmaker is mainly concerned with "making the books". This means that no matter what the outcome of a match is, the bookmaker should have earned a small profit on the bettings. In order to achieve this, a bookmaker will adjust the odds to the manner bets are placed on the game. Which team is actually more likely to win, does not matter, as long as for each possible outcome, the bookmaker will see a profit.

Given a team which has won 7 out of the last 10 games it has played, bettors will more often than not place their bets on a win for this team. As more and more people place bets on the same outcome, the bookmakers will adjust the odds for that outcome, by lowering them, thus reducing the payback received if the bettors were to win. If this is not enough to discourage more betting on this outcome, bookmakers will raise the odds of the other outcomes to make them more lucrative, giving bettors an incentive to bet on these results as well.

The bookmaker also reduces the odds, so that the expected return for a bettor will be less than 1 on average. If bets have been placed according to the following distribution (with regards to total amount of credit, not number of bets):

Home:60%
Draw: 15 %
Away:25%

The Bookmaker will tweak the odds to represent an e.g. 62% probability. Similarly, the probabilities for draw and away-win may increase from 15% to 16% and 25% to 27% , respectively. The table 1.1 below presents the consequences of this alteration.

Outcomes	Percetage of bets	Fair Odds	Presented odds	Expected Return
Home win	60%	1.67	1.61	0.97
Draw	15%	6.67	6.25	0.94
Away win	25%	4.00	3.70	0.93

Table 1.1: Betting distribution and odds presented by a bookmaker for a given football game

The thirld column of Table 1.1 shows how high the odds would have to be in order for the expected return to be exactly 1. As Table 1.1 shows, betting on an away win with the given bookmaker, you would receive 3.70 times the amount you placed, whereas if the 25% chance of away win is the real probability, you are receiving less than 93% of what you should have.

This means that if you were to place a bet of 100 credit on a a match with these odds each week, in the long run, you will loose 7% of the amount of credit placed each week. This 7% is how every bookmaker makes their profits (Though the actual percentage may vary).

Outcomes	Bets	Loss from outcome	Profits from outcome	Expected Return
Home win	60%	(1.61-1)*60=37	(15+25)=40	3
Draw	15%	(6.25-1)*15=79	(60+25)=85	6
Away win	25%	(3.70-1)*25=68	(60+15)=75	7

Table 1.2: Expected return for a bookmaker for a arbitrary football match, given the betting distribution described in column 2.

Table 1.2 shows how any bookmaker will have a positive expected return as long as, for each possible outcome, the expected return for a bettor is less than one. The exception is when all bets are placed on the same outcome. In this scenario, no matter how small the given odds are, the bookmaker stands to have a negative return should this outcome happen.

As we can see, if the bookmaker provides the odds presented in table 1.1, The bookmaker will have an average gain of slightly more than 5% of the total bets received for the match. This is usually the case, where the bookmaker has a profit margin ranging between 3 - 7%.

Now, consider an instance where the real probabilities are unknown (which they always are), but we can see the odds presented by the bookmaker. We know that the bookmaker has adjusted the odds according to the bets that have been placed before ours, and that these odds do not necesserily represent a good approximation of of the true probability distribution. If we can build a model which can accurately calculate the probabilities of each possible outcome of the game, we can now easily assess whether or not to place a bet on the given game.

If we assume our model is fairly accurate, and it presents us with the probabilities as presented in Table 1.3, we can now decide to either place bets on a draw, or an away-win.

Outcomes	Presented Odds	Model Probability	Expected Return
Home win	1.61	61%	0.98
Draw	6.25	11%	0.69
Away win	3.70	28%	1.04

Table 1.3: Probabilies presented by a bookmaker and by our model for a given football game

As we can see, the model predicts there is a 28% chance of an away-win. The odds should return a reward 3.57 times the size of our bet. However, since the bookmaker sees these outcomes as less likely, he is willing to pay back more than 3.70 times our placed bet should we win. If our model is correct, if you were to place a bet of 100 credit on an away-win in a match with these odds each week, in the long run, you will have gained 4% of the total amount of credit placed.

We are thus concerned with the possibilities of building a system that can be sufficiently accurate, and assess the probilities of match outcomes better than a given bookmaker. This in turn means that we only have to be better at betting than the outcome-distributions created by other bettors.

1.2 Research Questions

This section presents the research questions we wish to answer in this master thesis. This will be done by performing the experiments of chapter 6, and will be answered in the evaluation in chapter 7.

Research Question 1: Which in-depth data prove the most useful when attempting to improve the betting model?

We will be able to obtain several variables from each match played. Using all would require our model to sample a high amount of games in order to accurately describe the attacking and defending abilities of teams. It would therefore be of value to identify the most important variables so that our model can be kept as simple as possible, while at the same time using more information than only goals.

Research Question 2: Does using in-depth football match data improve our betting simulator?

Though it has been mentioned how including more variables should improve predictive models [21], it has not been tested in any research material we have been able to find. It will therefore be important to compare any model we produce with other models that *do not* use variables other than goals scored by each team, and assess whether the new model is an improvement on the old.

1.3 Research Method

The research method used for this project is mainly exploratory research, done by using the structured literature protocol given in Appendix B to produce knowledge of the problems concerned with predicting football outcomes. The structured literature protocol is built around finding relevent articles and papers on the subject by constructing query strings consisting of words and phrases assumed to be central to the research problem. Such phrases may be ”football”, ”betting”, ”model”, etc. The set of strings produced are then all permutations of these phrases together. The strings are then used as search queries on databases and other types of sources known to have relevent articles within the domain of prediction.

When the relevent articles are found, a review of the different research done is provided, listing the important aspects of each study, and comparing them on these bases.

There are also two constructive research stages, where first a set of older models are tested and compared, and then secondly our proposed model is compared to the models in the first test, using the same datasets and betting methods, to assess which provide the best results.

2 Background Theory and Motivation

This chapter summarizes the Structured Literature Review first presented in the preliminary study for this thesis, Ellefsrød(2012), after first introducing the statistical tehniques required to appreciate those results. The Structured Literature Review is reproduced in its entirety in Appendix B.

2.1 Background Theory

This section provides an introduction to some of the theory used in existing research, as well as that used in this project. Most importantly, we here present Gibbs sampling, a Markov chain Monte Carlo algorithm used for Bayesian inference of latent parameters in the model. We will also introduce some of the technology mentioned in this report.

2.1.1 Markov Chains

For a given sequence of random variables, $\{ V_0, V_1, V_2, ... \}$, given that the next state V_{t+1} for each time $t \geq 0$ is only dependent on the previous state V_t, we say that this sequence is a *Markov chain*. That is, the next state V_{t+1} is sampled from the distribution $P(V_{t+1}|V_t)$, which depends on only the current state of the chain, V_t [22]. More formally,

$V_{t+1} \perp\!\!\!\perp \{V_0, V_1, V_2, ..., V_{t-1}\}|V_t$.

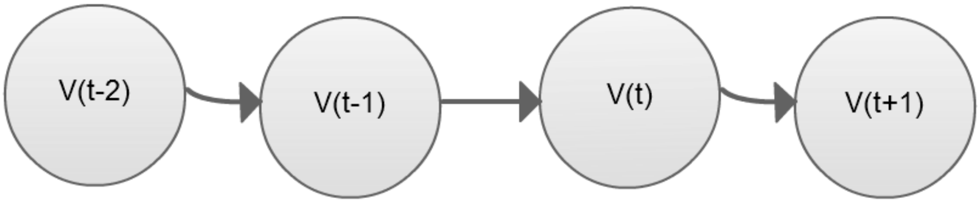


Figure 2.1: Presenting the conditional independence assumption of the Markov chain by using a simple Baysesian network.

Figure 2.1 shows a Markov chain, which follows the *Markov assumption*, which states that every state provides enough information to make the future conditionally independent of the past. We will be utilizing this when assessing attacking and defending abilities of teams later in this article.

2.1.2 Gibbs Sampling

Gibbs sampling is a Markov chain Monte Carlo algorithm used for statistical inference of one or more model parameters, or missing data. This could for instance be determining the the population of a city by counting number of cars passing a point on a road on a given day. In the case of determining strengths of football teams, Gibbs sampling may be used in determining parameters such as attacking and defending strengths, by observing how many goals a team scores on average when playing home games, or when playing away games, or it could be how many shots a team gets on target during a game.

Gibbs sampling provides a method for obtaining approximations of the marginal distributions of the variables we are interested in. For a given vector $\boldsymbol{X}$ of k components, we wish to obtain n samples for this vector. These n samples are then used to construct the posterior distribution for each component. In the models presented in later chapters, these components encompass, amongst others, model parameters such as attack and defence strengths of each team, and missing data values such as goals scored, passes made, shots taken, etc.

Gibbs sampling starts off with some random value for each variable $j = \{1...k\}$ in the vector $\boldsymbol{X}_0$. At each time t, the next state $\boldsymbol{X}_{t+1}$ is chosen by first sampling a candidate point $\boldsymbol{Y}$ from a proposal distribution $q(.|\boldsymbol{X}_t)$. For the Metropolis-Hastings algorithm, the more general form of Gibbs sampling, the candidate point is then accepted by the probability $\alpha(\boldsymbol{X}_t, \boldsymbol{Y})$, where

$\boldsymbol{X}_t$

$$\alpha(\boldsymbol{X}, \boldsymbol{Y}) = \min\left(1, \frac{\pi(\boldsymbol{Y})q(\boldsymbol{X}|\boldsymbol{Y})}{\pi(\boldsymbol{Y})q(\boldsymbol{Y}|\boldsymbol{X})}\right)$$

and $\pi(.)$ is the distribution of the k variables in $\boldsymbol{X}$ [8]. The Gibbs sampler, however, chooses proposal distributions q such that

$$\frac{\pi(\boldsymbol{Y})q(\boldsymbol{X}|\boldsymbol{Y})}{\pi(\boldsymbol{Y})q(\boldsymbol{Y}|\boldsymbol{X})} = 1$$

,

ensuring that each new candidate point is always accepted. Instead of updating the whole of $\boldsymbol{X}$ at once, it is both more convenient and efficient to divide$\boldsymbol{X}$ into its components $\{\boldsymbol{X}_{.1}, \boldsymbol{X}_{.2}, \boldsymbol{X}_{.3}, ..., \boldsymbol{X}_{.k}\}$, and update each one by one. The matter then becomes to generate a candidate $\boldsymbol{Y}_{.j}$ for each of the k variables in $\boldsymbol{X}$.

For each sample $t = \{1...n\}$, we can sample a variable j in $\boldsymbol{X}$ from the distribution of j conditioned on all other variables, using the most recent values of the other variables in $\boldsymbol{X}$, and updating j when it has been sampled:

$$\boldsymbol{X}_{t.j} = p(\boldsymbol{X}_j | \boldsymbol{X}_{t.1}, \boldsymbol{X}_{t.2}, ..., \boldsymbol{X}_{t.j-1}, \boldsymbol{X}_{t-1.j+1}, ..., \boldsymbol{X}_{t-1.k}$$

Together, the obtained samples then approximate the joint distribution of all k variables, while looking at the sampling of a single variable j provides an approximation of the marginal distribution for that variable. For Bayesian applications, the vector $\boldsymbol{X}_t$ will contain both model parameters and missing data.

In the case of infering goals scored by a team in a given match, if we had the previous match result available, we would then use the amount of goals scored in that match to provide a sample for this one.
Gibbs sampling generates a Markov chain of samples, and each sample is correlated with the ones that came imidiately before and after it. As we are interested in independent samples, there are two important features of Gibbs sampling we must take note of:

Thinning: Since there is a correlation between neighbouring nodes in the generated Markov chain, it is important to thin the chain by only using every ith value. Having the i sufficiently large, the apparent dependency between nodes (samples) in the chain will be negligible. Having the correlation between samples low helps the Gibbs sampler take larger 'steps' between each sample, giving a better approximation of the marginal distributions of each component of $\boldsymbol{X}$ in smaller sample sizes.
There exists a pay-off between computational cost and obtaining near-zero independence, as increasing n would make the algorithm run longer, but the results would be more accurate.

Burn-in: At the beginning of the chain, it is often the case that the samples do not accurately represent the desired distribution, and it may take some steps before the chain does so. The burn-in period defines how many samples we produce at the start of the chain, before we start keeping them.

We should also take note that Gibbs sampling is a randomized algorithm, and hence may produce different results each time it is run. Building a large Markov chain with many samples, together with sufficiently high thinning and burn-in parameters help some way in ensuring that the difference in each running of the algorithm stays small. Assuming this, we may save a particular run of the Gibbs sampler and reuse the results for testing different betting strategies on a specific model. This is a very important feature to mention, as it will save much time. Depending on how many samples, how large thinning and how much burn-in, as well as considering if parallell computing is utilized or not, it may take several hours to simulate a whole football season's results.

2.2 Assessment of Maher, Dixon & Coles

This section presents the contributions Maher (1982) and Dixon & Coles (1997) have made into the field of football prediction and team strength assessments. This is a more compact reiteration of the background research fully presented in Appendix B, but is highlighted in this section for the readers benifit.

2.2.1 Using the Poisson Distribution

Maher (1982) describes how early work in the field of modelling team strengths and predicting football match outcomes used the negative binomial distribution to model the amount of goals a team would score during a given match. Maher (1982) states that this assumption erroneously implies that all teams have equal strengths, contradicted by works that show how football league final standings may be quite correctly predicted by experts. This seems a strong indication that Maher (1982) may be right in his assumption that chance may play a considerable role in a single match, but over several matches it dissipates, overshadowed by the differences in team abilities.

Maher (1982) assumes that each time a team has possession of the ball, it has the opportunity to attack, which may subsequently result in a goal. With n attacks during a game, with the probability p of an attack resulting in a goal, the number of goals can be approximated by the Poisson distribution. This requires the assumption that:

1) the probabiltiy p for a goal is constant in each attack and

2) the outcome of each attack is independent of any other attacks.

Maher (1982) provides an interpretation of a football match which gives rise to a binomial probability of number of goals, approximated by the Poisson distribution. Though using the Poisson distribution has provided many good results, it may be questioned whether these are reasonable assumptions. One may argue that an attack starting with winning possession of the ball from the opposition goalkeeper yields a higher p value than an attack starting with tackling the opposition striker in your own penalty-area. An attack leading to one team going into a 3-0 lead may have negative effects on both teams' p value, or even n, as securing the win by keeping possession of the ball becomes a higher priority, leading to fewer attacking opportunities for both teams.

2.2.2 Introducing the bivariate Poisson model

Utilizing the Poisson distribution, Maher (1982) uses the product of the home-team's attack ability and away-team's defending ability as the mean amount of goals scored by the home-team, and vice versa for the away'team. Maher (1982) first assumes the goals scored by each team are independent of each other, before improving his model by introducing a bivariate Poisson model. When using the independent models for home- and away-goals, Maher (1982) states that they can be interpreted as two separate games at each end of the pitch, which may be a over-simplification of the game. This is also demonstrated by the bivariate Poisson version of the model, which improves the results considerably.

The correlation between goal scoring at the two ends of the pitch has been debated in several papers. Karlis & Ntzoufras (2003) also considered, as Maher, using the bivariate Poisson distribution to model team capabilities, and shows how increasing the correlation between goals scored, has a positive effect on the prediction of amount of games drawn. Using a correlation factor of 0.2, as Maher (1982) used, gives a 14% increase in expected number of draws. Karlis & Ntzoufras (2003) further improve their model by using similar inflating methods as Rue & Salvesen (2000) and Dixon & Coles (1997), but rather than inflating the results 0-0 and 1-1, Karlis & Ntzoufras (2003) inflate the probabilities of draws in general. The bivariate Poisson model gives more expressiveness. With the independent model, assessments such as "team A tend to win 1-0" can be made, whereas the bivariate model provide the means to assert that "team A tend to beat team B by 2 goals", because the amount of goals scored by each are now correlated.

2.2.3 The Home Ground Advantage and Varying Team Form

By testing increases in Maximum Log Likelihoods when adding parameters to his model, Maher (1982) found that introducing individual attacking and defending abilities increased the accuracy of his model. Meanwhile, adding parameters to describe a team's strengths when playing away games did not increase the likelihood of the model at the 1% level, and Maher (1982) concluded that it is enough to add a constant factor for all teams to provide for the advantage that comes with playing at home. Maher (1982) does not provide any details on the origins of the home-advantage effect and whether or not it is a fair assumption, but Dixon & Coles (1997) show that for the period 1993-1995 and over 6000 matches in English football, the ratio of outcomes are 46% home wins, 27% draws and 27% away wins, which provide enough evidence for this to be a valid assumption to make. Cattelan et al. (2012) also showed that for the 2008-2009 season in the Italian Serie A, an average of 65% of points each team accumulated over the season, was obtained in home-games. Knorr-Held (2000) provides data from the 1996-1997 season of the German Bundesliga, showing that of all the games played, 51% ended in home wins, and only 26% resulted in away wins. Put in context to each other, these findings strongly indicate that whichever footballing league one uses data from, and whenever these data are from, there seems to be an inherent advantage to the team playing at home, for whichever reason.

Where Maher (1982) assumed that each team's strength was constant over the period of a season, several others have later attempted to use dynamic attacking and defending abilities of teams to capture the variable performances a given team may have over the season. Though this may seem a subjective opinion, and that variations in a team's (superficial) performance may be caused by chance, there are several reasons why chance may not have the only say in this: Dixon & Coles (1997) and Knorr-Held (2000) state how performance in a particular game may be influenced by the ability of newly arrived players, changing of the coach, unavailability of injured players or sacking of a manager. This seems reasonable, as removing or including an essential part of a team may easily alter the overall strength of that team. One must however also consider that adding the possibility of variable team strengths may lead to an overfitting of the model, where a few wins followed by a couple of losses leads to the team being interpreted as first one of the best teams in it's league, then quickly reduced to one of the worst. Rue & Salvesen (2000) use a parameter t to indicate how far back in time we will look to find match results used to estimate the team's current strength. Each match then has a decreasing influence on the team's current ability, as we move further away from it in time.

2.2.4 Altering the Poisson Distribution by Inflation

Dixon & Coles (1997) build upon the approach taken by Maher (1982), introducing a simple approach to such a fluctuation in team capabilities. They too stick to using each team's history of match scores alone, in order to estimate strengths. Rather than building on the conclusion of Maher(1982) that the bivariate Poisson model provided better results, Dixon & Coles (1997) use the initial independent assumption of goals by the two opposing teams. They find that the independent model is particularly bad at predicting the scorelines 0-0, 1-0, 0-1 and 1-1, and that the bivariate Poisson distribution does not sufficiently improve these results. Hence the model is modified to improve the expected amount of the mentioned four outcomes, while keeping the marginal distributions of goals scored by teams X and Y Poisson. There are however more score-lines which the Poisson distribution either over- or under-estimates; 4-3, 3-4, 3-3, and 6-1 are all results which suggest the independence between scores is unreasonable, but the modified Dixon & Coles (1997) model does not take these into account. One may argue that these results are relatively rare compared to those that the authors adjust for, and because of this, it takes only a few occations too many or few in the sample set to make the model seem unreasonable.

Rue & Salvesen (2000) use Bayesian methods to update time-dependent estimates of team strengths each time a new match has been played, and the Markov chain Monte Carlo (MCMC) techniques are iteratively used for inference of simultaneous, dependent abilities of all teams in a league.

2.2.5 Comparision of Models

The different models proposed are difficult to compare as there is a vast sample space of data which a researcher may use for building their model, and it only continues to grow as time passes and more football matches are played. Data used varies between the years 1970 to 2007, and different leagues have also been used, such as English, German and Italian. It is then problematic to assess whether one model has classified the strength of Bayern Munchen in the German Bundesliga of the 1996-1997 season in a better manner than someone else classified Manchester United in the English Premiership in 2006-2007. This may be easierly done when the researchers proceed to use these models in a betting environment to predict future matches.

Not all of the research done provide empirical tests of models, where the model has been used to predict matches, and been applied to betting strategies. Rue & Salvesen (2000) provide

a betting strategy of betting on outcomes with positive expected profit, while at the same time keep the variance in profit low. They also attempt combination betting, where they tried predicting three matches at a time. This proved less successfull, but they managed to get a profit when placing single bets, though the lower bound of the variance in the results indicated there was still some risk in losing money. Dixon & Coles (1997) also attempted a betting strategy with their model, and provide results which are borderline significantly larger than the return expected with random betting coupled with the standard bookmaker's take. The variance is, however, as in the case with Rue & Salvesen (2000), very large, and a definite conclusion is difficult to make.
Maher (1982) is more interested in examining how well his model predicts the number of goals in matches, which it does quite well, than predicting actual outcomes in matches. For instance, he examines the count of expected number of matches in which team A scores 1 or 2 goals, or team B scores 1 or two goals, or the difference in goals is -1, 0 or +1, and compares these to the observed counts of such events. He does not, however, attempt to examine if there is an overlap in observed and expected events, for instance; did a game that ended 2-2 also be predicted as 2-2, or a draw?

As no conclusions can be drawn, the next chapter will compare the Dixon & Coles (1997) and Maher (1982) models using data from the English Premier League, season 2011/12, and try to assess which of the models give the best results.

2.3 Technologies and Workflow

In this section we give a description of technology used in order to execute the experiments in the next chapter. We will also go through the details of the workflow during this project.

2.3.1 MATLAB

MATLAB, or *Matrix Laboratory*, is a high-level programming language as well as a numerical computing environment. MATLAB is especially effiecient and easy to use with regards to mathematical problems and algorithms, and vizualization in terms of graphs and diagrams. Because of the extensive build-in library of mathematical functions, MATLAB is better suited than traditional programming languages, allowing us to reach solutions faster and easier. MATLAB also contains functions for integrating with Java or .NET, making it possible to utilize the functions MATLAB provide in more extensive solutions. However, MATLAB does not contain any built in functions for doing Gibbs sampling.

2.3.2 JAGS

Just Another Gibbs Sampler, is a program for analyzing Bayesian models using Gibbs sampling. It provides no graphical user interface for building models or postprocessing samples, and must therefore be used in tandem with a separate program. R is one such possibility, MATLAB is another. We will use the latter option.

JAGS modelling is done using a dialect of the BUGS language, which is also used in WinBUGS and OpenBUGS, too. Both WinBUGS and OpenBUGS are programs for Bayesian analysis, where WinBugs was for the Windows operating system mainly, and OpenBUGS has more operating system options as well as being licensed under the GNU General Public License.

2.3.3 HTML

HyperText Markup Language is the main language used for creating web pages. HTML uses both predefined and user-defined tags to describe how the content of a webpage should be displayed. A browser will read HTML documents and present the content according to how the elements of the document are labled, i.e. how the structure of the HTML tree is.

We will mostly be encountering HTML-structured text when searching through source code of web-pages.

2.3.4 PHP

PHP, a hypertext preprocessor mainly used for creating dynamic webpages, is a dynamic and loosely typed programming language. PHP is used for development on the server-side of an application, abstracted away when seen from any client. When loading a web-page, any PHP code will be run by the server before an application is run by the client [19].

Because an open-source PHP-script for parsing HTML exists, i.e. the *simple_html_dom.php* script, we will use PHP mainly for retrieving source code from webpages that present desirable information that we wish to organize in our own system. PHP has built-in pattern-matching functions, which we will use for seeking out specific elements in source code.

PHP also allows us to manipulate data-files, as long as we have access to those files (which we do as long as we are operating from the server itself).

2.3.5 WampServer

WampServer is an open source, Windows web development environment. It allows us to build web applications with, most significantly for us, PHP [23]. Though our service will not be intended to be accessed from other users on the web (as we will be using internet browsers mainly as an easy interface to our crawler functions. These will be altering .csv files on the server, which other external clients of the application will not have access to), we will need to set up a a web server in order to ensure that a browser can interpret the PHP code. WampServer is easy to set up and will automatically install the requirements needed in order for us to start developing. `http://www.wampserver.com/en/#download-wrapper` provides an install guide, and section C of the Appendix gives a short description of how to circumvent other applications using the default WampServer port.

2.3.6 Regular Expressions

A regular expression (or *regex*) is a sequence of characters; some characters may have their literal meaning while yet some may be metacharacters that represent a set of possible characters. Regular expressions are used when searching through text to identify a subset of the text that we wish to do further processing. In some cases we know the exact form of the text we are looking for, while in others we only have a general notion of what the text should look like. It is in the latter cases that regular expressions come in handy.

Some of the meta characters are described in table 2.1. We will be using regular expressions to identify match-ids, team names, attributes, etc. in source code retrieved from web-pages.

Character	Meaning
.	Any character except newline.
\.	A period (and so on for *, \(, \\, etc.)
$	The end of the string
\d, \w, \s	A digit, word character [A-Za-z0-9_] or whitespace.
[abc]	a.
aa|bb	Either aa or bb.
+	One or more of the preceding element.
?	Zero or one of the preceding element.
*	Zero or more of the preceding element.
{m,n}	Between m and n of the preceding element.

Table 2.1: List of some of the prominent metacharacters of regular expressions

2.3.7 Workflow

The time during the preliminary period of the project went into producing the Structured Literature Protocol found in Appendix B, finding sources and examining and documenting state-of-the-art methods. A MATLAB framework for using JAGS models was also provided during this period, and time was spent learning the framework, as well as the fundemental parts of the MATLAB language and programming environment. the BUGS language used in JAGS for creating models was also researched for later development of own models.

With the MATLAB framework there was also provided the JAGS-implementation of the Maher(1982) model. Beyond this, the Dixon & Coles (1997) model has been implemented as well. Both these models have then been tested to produce empirical evidence of their accuracies, presented in the next chapter. A description of the Dixon & Coles (1997) model can be found in Appendix A.

During this project an *internet crawler* has been developed, described in the following chapter and in further detail in appendix B. The internet crawler gets information from specific web-sites to that we are able to build a data-set which includes more extensive data from each match, such as shots made, passes, possession statistics, tackles made, etc. This was a core element needed to be done in this project in order for us to assess whether adding statistics beyond goals scored and looking behind the results would improve model accuracy and betting score, compared to the models of Dixon & Coles (1997) and Maher (1982).

Two main models using match statistics have been developed in JAGS, and the framework has been extended to allow the inclusion of more variables for matches, as well as statistical displaying functions for comparing betting models and reviewing variables in general. This could for instance be finding the error of a linear regression line when plotting the correlation between goals scored and shots made.

3 System Architecture

This chapter gives a detailed description of the architecture of our simulator which will be used for testing different betting models. We will begin by describing the information-gathering part, which gathers data about football matches after and before they have been played, and the simulator itself.

3.1 The internet Crawler

Each week, a new round is completed in the English Premier League, working it's way towards the 38th and last round of the season. We want to make a simulator which can at any given time utilize the results and data of all the games that have been played up until that point in time, in order for us to have a best possible foundation for the predictions we are about to make. In order for it to be interesting, we would also like the bookmakers odds on the upcoming matches which will be played next, so that we can have something to compare our computed probabilities with.

Taking advantage of the *simple_html_dom* -PHP script, we can retreive the HTML and javascript source code of any webpage on the internet, and using regular expressions we can specify which part of that code we are interested in [18]. In our solution, we will use this to retreive:

1) Match data from `www.whoScored.com`, and

2) Match odds from `www.betexplorer.com`.

This section will focus on explaining how this is done, and how the information is stored in the system. All code apart from the *simple_html_dom.php* -script was written by the author of this document, and no other external code was used.

3.1.1 Important Data Files

This section describes some of the files used for storage of data, both intermediate data and final data used in the simulator. Further examples of the data-files can be found in Appendix C, where there are more detailed descriptions of how the most important PHP-functions work. Figure 3.1 gives an overview of the files mentioned in this section.

1) rawTable.txt Holds the initial, unaltered source code we retreive from getting the league table off of the whoscored.com main page for the league we are interested in. RawTable.txt contains a part of a javascript function-call used for setting up the current league table. Arguments for this function are league-ID,team-IDs,team-names, and HTML hyperlink-snippets for each of the 12 most recently played matches for each team. An example of how this file looks like can be seen in Appendix C.

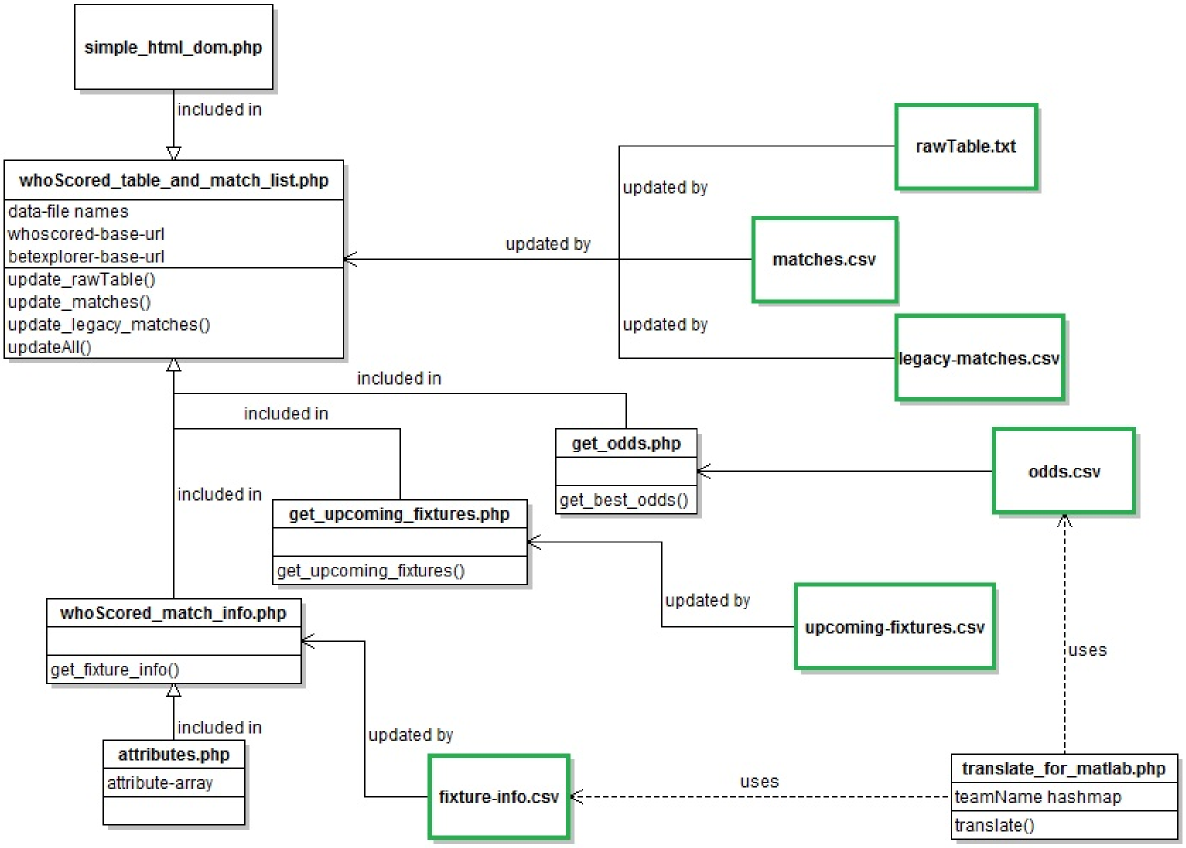


Figure 3.1: An overview of the structure of the crawler.

2) matches.csv Each HTML hyperlink contained in rawTable.txt contains an ID for the match it represents, which is used in unique URL for that game's info page. Using regular expressions and pattern matching, we identify all these IDs in rawTable.txt, and which teams participated in each match. Matches.csv contain a list of each team, together with the IDs of the last 12 matches each team has played, presented as comma-separated values. Below is an example of how this file may look like:

```
32,'Manchester United',615224,615270,615278,615298,615303,614137,615207,615228
167,'Manchester City',615275,615282,615297,615262,615306,614132,615199,615243
15,'Chelsea',615269,615280,615298,615260,615300,614129,615203,615226,615240
13,'Arsenal',615214,615268,615278,615292,615293,614133,615193,615230,615258
```

```
30,'Tottenham',615275,615291,615299,615260,615307,614136,615212,615236,615256
31,'Everton',615214,615273,615281,615296,615301,614129,615204,615227,615234
```

The first value on each line represents the ID of the team, followed by the teamname, and lastly the IDs of the matches. As can be seen from the example, the same ID (615260) is registered twice, as both Chelsea and Tottenham played this game.

3) legacy-matches.csv As matches.csv only contains the IDs of the last 12 matches, we need to save these match-IDs somewhere before updating matches.csv. This is done in legacy-matches.csv, and has the exact format as matches.csv, except each team has registered up to 38 match-IDs.

4) fixture-info.csv Having obtained the match-IDs, we now know the URL of each match, and therefore we can obtain the source code from each match-page on the `whoScored.com` domain. Using regular expressions and pattern matching, we are able to find the match-info. This is then stored in fixture-info.csv, where each line represents a single fixture. Headers in the first line indicate what each value represents. An example (of a single line) follows below:

```
614052,31,32,'Everton','Manchester United','08/20/2012 20:00:00',
'1 :  0',2,9,7,2,18,7,13,18,19,0,6,275,196,30,18,28,1,4,6,4,0,
14,7,11,23,15,1,8,646,571,69,28,18,0
```

The first 3 values indicate the match-, hometeam- and awayteam-IDs, and the last 34 values are 17 different variables of performance for each team. These variables are given in table 5.1

5) upcoming-fixtures.csv The main page for each league at `whoScored.com`, in addition to containing a league table, also has a list of fixtures for the current month we are in. upcoming-fixtures.csv contains the fixtures of this month that have not yet been played, and each line is of the form:

```
614052,31,32,'Everton','Manchester United','08/20/2012'
```

6) odds.csv Values from each line in upcoming-fixtures.csv are used in a regular expression for pattern-matching when crawling `www.betExplorer.com`. For each match, we obtain the best odds for each outcome, along with the name of the bookmaker that provides that odds. An example line from odds.csv may look like:

```
1,Sunderland,Arsenal,MarathonBet,4.75,bet365,3.75,William Hill,1.95
```

The order of the odds is home-win, draw, away-win, i.e. MarathonBet provides that best odds for a home-win, William Hill the best for away-win, and bet365 the best for draw. The first value in the example indicates whether we have obtained the newest odds for the match in question. If the value is 1, then we need not update. If the value is 0, it means that we have recently tried to update the odds for upcoming matches, but for some reason couldn't update them all. The lines prefixed with zeroes indicate that this is where we should start updating the next time. The most prominent reason for such zeros to occur is that it takes time to download the source of around 20 web-pages, and a browser will time out after 30 seconds. Thus we stop after a predefined maximum limit, and update the odds we managed to obtain in that time-frame.

3.1.2 Important PHP-scripts

This section goes through the PHP-scripts used by our information-gathering crawler, describing the php-scripts and which tasks each perform.

1) simple_html_dom.php This is an open-source script licensed under the MIT License, which enables us to easily load source code of webpages [18]. Example use of this script can be found in Appendix C.

2) whoscored_league_table_and_match_list.php This is the main script, and includes all the other scripts, either directly or indirectly. It also holds the global variables that need to be changed in order for us to obtain data for another league, or another season. A description on how to do this can be found in Appendix C.

This script contains the method used for updating the rawTable.txt file, as well as updating the matches.csv file. It also has a general function updateAll() that updates the data-files in the following order: rawTable.txt - matches.csv - legacy-matches.csv - fixture-info.csv - upcoming-fixtures.csv.

3) whoscored_match_info.php This script contains functions for updating the fixture-info.csv file. It also contains helper functions for ensuring that the program does not stop while updating the data-file, and also for not spending unnecessary time crawling matches that data already have been obtained from.

4) attributes.php contains a list of attributes represented as strings, which are used to compare with the values obtained with the crawler for an arbitrary match. These attributes are values such as blocked scoring attempts, shots, passes, possession, goals, etc.

5) get_upcoming_fixtures.php Contains functions for retrieving a list of upcoming matches in the current month, and stores this list in the upcoming-fixtures.csv file. Also cotains helper-functions for transforming date format for easier use in MATLAB.

6) get_odds.php Contains functions for retrieving odds from `www.betExplorer.com`, and helper functions for sorting different bookies and selecting the ones which give the best odds for a given match.

This script also contains an important hashtable for translating team names; we will be using names used from upcoming-fixtures.csv, which was obtained from `www.whoScored.com`, in the pattern matching algorithm for finding odds. These two sites have slightly different naming conventions, such as one site using the name *West Bromwich Albion* and the other simply *West Brom.*

7) translate_for_matlab.php This script is used for concatenating the two data files odds.csv and fixture-info.csv, and saving it to a third file. This file is the one converted into a .MAT data file used by the simulator.

3.1.3 Taking it step by step

There are several steps we must go through in order to end up with the complete match data for every match that has been played so far in the league, containing all the in-game statistics and also the odds for all three possible outcomes provided by bookies. Our system solves this problem by systematically going through the procedure described below:

Finding the page-URLs for each match where the wanted information is presented and retrievable. Each match is summed up on unique pages under the `www.whoScored.com` domain, and for each match, their URL is distinguished from all other matches by the site's use of unique match IDs.

For example, the URLs for the two webpages presenting data from the matches Liverpool - Queens Park Rangers and West Bromwich Albion - Manchester United (two matches taken from the final round of the season), are:

```
http://www.whoscored.com/Matches/614130/Live
http://www.whoscored.com/Matches/614137/Live
```

This problem is then reduced to identifying the unique IDs of all the matches in the Premier League. This is done by utilizing the front page of the league we are interested in. In our case this is `http://www.whoscored.com/Regions/252/Tournaments/2/Seasons/3389`. Here is found a table of the current league standings, as shown in Figure 3.2. As we can see, this table contains a form column, showing how each team has fared the last 6 matches. At the top of the table we also see that a viewer can distinguish between *overall*, *home* and *away* views of the table. Since home and away matches for each team are mutually exclusive, this table will contain match-IDs of the latest 12 (at most, less if we have not yet come that far into the season) matches played by each team.

Using the regex matching technique described in Appendix C, we use the pattern

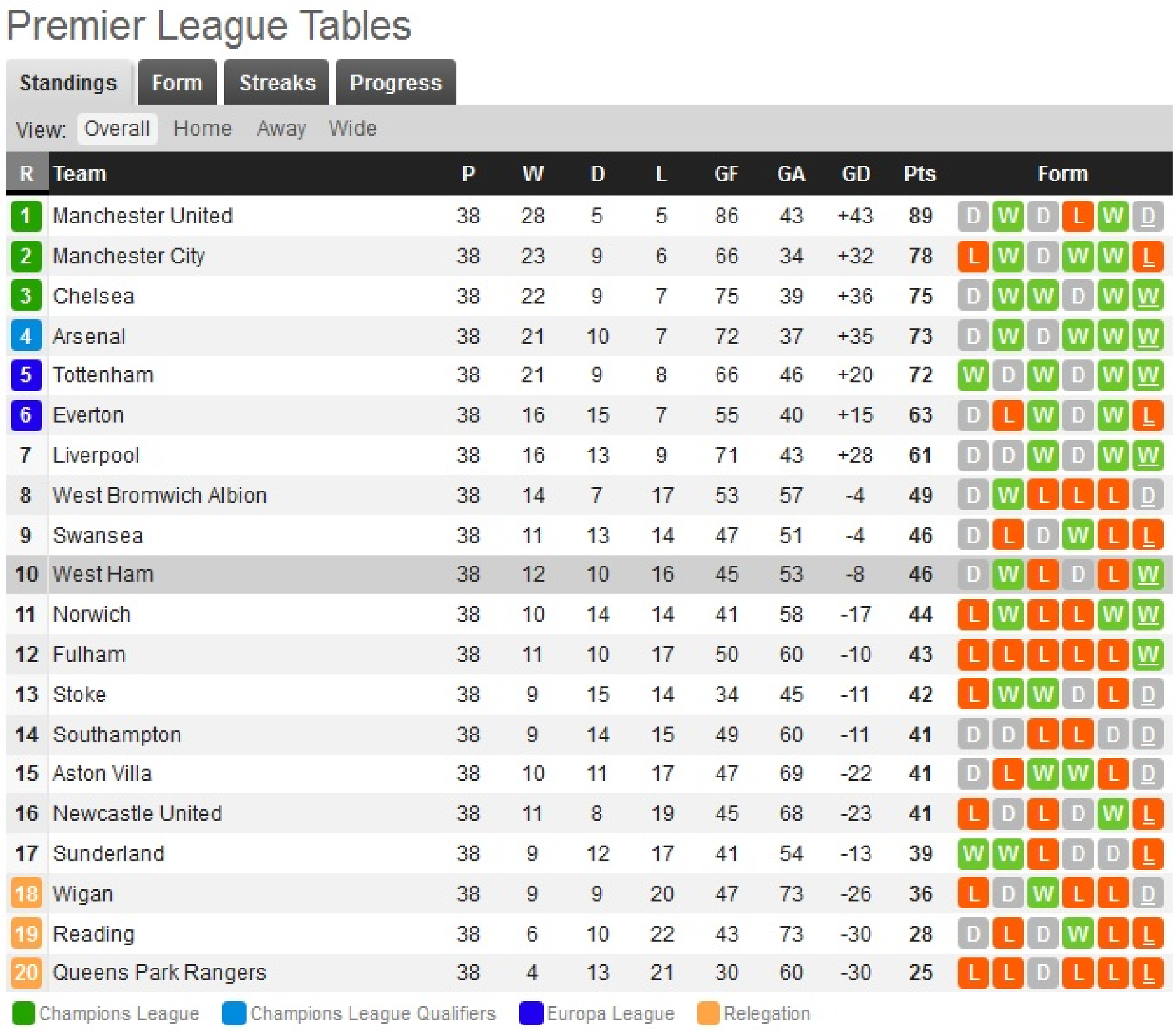
```
"/Datastore\.prime\(’standings’, { stageId:  ".$stageID."}, \[(\[.*\n,?)+/"
```

and find the source code for the table. An example of how this looks like is given in appendix C. The next step is to further extract each unique matchID from the table-source code. For this, a much less complicated pattern is sufficient, because we know that each match's ID-tag is inside an HTML hyperlink, and each hyperlink uses the match-ID as an attribute. For example, the following may be a hyperlink contained in the line containing fixtures Arsenal are involved in:

```
<a class="d h" id="615214" title="Arsenal 0-0 Everton"/>
```

The pattern below will extract any string that starts with ' id=" ' , followed by at least one digit, ending with a ' " '. These are then stored in the matches.csv and legacy-matches.csv files, and we have enough information to now know the URLs where we will find the match statistics for all the fixtures we are interested in.

Pattern for finding id-tags: `"/id="[0-9]+"/"`



Premier League Tables

R	Team	P	W	D	L	GF	GA	GD	Pts	Form
1	Manchester United	38	28	5	5	86	43	+43	89	D W D L W D
2	Manchester City	38	23	9	6	66	34	+32	78	L W D W W L
3	Chelsea	38	22	9	7	75	39	+36	75	D W W D W W
4	Arsenal	38	21	10	7	72	37	+35	73	D W D W W W
5	Tottenham	38	21	9	8	66	46	+20	72	W D W D W W
6	Everton	38	16	15	7	55	40	+15	63	D L W D W L
7	Liverpool	38	16	13	9	71	43	+28	61	D D W D W W
8	West Bromwich Albion	38	14	7	17	53	57	-4	49	D W L L L D
9	Swansea	38	11	13	14	47	51	-4	46	D L D W L L
10	West Ham	38	12	10	16	45	53	-8	46	D W L D L W
11	Norwich	38	10	14	14	41	58	-17	44	L W L L W W
12	Fulham	38	11	10	17	50	60	-10	43	L L L L L W
13	Stoke	38	9	15	14	34	45	-11	42	L W W D L D
14	Southampton	38	9	14	15	49	60	-11	41	D D L L D D
15	Aston Villa	38	10	11	17	47	69	-22	41	D L W W L D
16	Newcastle United	38	11	8	19	45	68	-23	41	L D L D W L
17	Sunderland	38	9	12	17	41	54	-13	39	W W L D D L
18	Wigan	38	9	9	20	47	73	-26	36	L D W L L D
19	Reading	38	6	10	22	43	73	-30	28	D L D W L L
20	Queens Park Rangers	38	4	13	21	30	60	-30	25	L L D L L L

🟩 Champions League 🟦 Champions League Qualifiers 🟪 Europa League 🟧 Relegation

Figure 3.2: League table taken from www.whoscored.com

Retreiving match information is the next step. For each team, match-IDs are extracted from legacy-matches.csv for completing URLs. Knowing now the URL of the webpage that holds the match-statistics we are looking for, we use the following procedure to find the part of the source code where that information resides:

- We need to find the names of the two teams involved in the match, in order to use them in creating a new pattern for finding the statistics. The following line is a Javascript function-call pulled from the source code of the page, and it's arguments provide us with the information we need.

  matchHeader.load([13,194,'Arsenal','Wigan','05/14/13',6,'FT','1 : 1','4 : 1',,,'4 : 1']

- The ID-tag and name of each respective team are then used to find the statistics we are interested in. The following snippet is the matching string we find with the pattern `" \[\[$info[0], $info[2],.* \] \] \]\], \[ \[ /"` , where `$info[0]` is Arsenals team-ID 13, and `$info[2]` is the string 'Arsenal':

  ```
  [[13,'Arsenal',7.29,[[['blocked_scoring_att',[4]],['att_miss_right',[1]]
  ,['att_goal_low_left',[2]],['accurate_pass',[363]],['att_goal_high_centre
  ',[1]],['att_miss_left',[4]],['total_tackle',[24]],['total_offside',[2]]
  ,['att_sv_low_left',[2]],['att_goal_high_right',[1]],['att_sv_low_centre',
  [3]],['won_contest',[5]],['att_sv_high_centre',[1]],['shot_off_target',[7
  ]],['ontarget_scoring_att',[10]],['total_scoring_att',[21]],['aerial_lost
  ',[10]],['fk_foul_lost',[12]],['total_throws',[23]],['won_corners',[7]],
  ['possession_percentage',[47.5]],['aerial_won',[12]],['total_pass',
  [452]],['att_miss_high_right',[2]],['goals',[4]]]],
  ```

- It is then a matter of splitting the string on commas, and trimming away all excessive brackets. The variables are then placed in an array and compared to the values in *attributes.php*. This is an important part, because the values in the snippet are not ordered in any specific way. Also, if an event has not occured, such as having a shot blocked, that event will not be listed at all, instead of being listed as `['blocked_scoring_att', [0]]`. This will make our data-files skew, and also having different variables in the same columns. We wish to keep our files rigid, and therefore we are only interested with the variables found in *attributes.php*.

Retreiving upcoming matches We next want to update *upcoming-fixtures.csv*, which contains the match-ID, team-IDs and names for fixtures coming up. This will then be used for getting bookie odds for matches that have not yet been played, so that we can use the simulator to place real bets on matches if we chose to.

Using the pattern `"/DataStore.prime \('stagefixtures',.* \n(,.* \n)* \] \);/` we retrieve the following code snippet:

```
DataStore.prime('stagefixtures', $.extend({ stageId: 6531, isAggregate: false
}, calendarParameter),
```

```
[[615287,1,’Saturday, May 4 2013’,’15:00’,175,’West Bromwich Albion’,0,194,’Wigan’,0,’2
:  3’,’1 :  1’,1,1,’FT’,’2’,0,1,1,0]
,[615289,1,’Saturday, May 4 2013’,’15:00’,168,’Norwich’,0,24,’Aston Villa’,0,’1
:  2’,’0 :  0’,1,1,’FT’,’2’,0,1,1,0]
...
,[614132,1,’Sunday, May 19 2013’,’16:00’,167,’Manchester City’,0,168,’Norwich’,0,,,0,0,’
]);
```

The sixth to last value on every match shown here is important. It has been observed to have 5 distinct values;

- **Positive digit between 1-90.** This indicates the game is currently being played.

- **’HT’** The game is currently at the half time break.

- **’FT’** The game has been completed.

- **-1** The game has not yet started.

- **’Postponed’** The game has been postponed to a later time.

We are looking to find upcoming matches, and so all matches that do not contain a ’-1’ in this position, are ignored. This includes postponed matches, because we do not know how far into the future the new match date will be set. As we will be using these match-IDs for finding odds on `www.betExplorer.com`, postponed matches may not yet have received odds.

Finding best bookmaker odds for upcoming matches: `www.betExplorer.com` presents on their main page for the English Premier League, `www.betExplorer.com/soccer/england/premier-league`, the average bookie odds for upcoming matches. This can be seen in Figure 3.3.

We wish to obtain the best possible odds for a fixture available at any given time. We must then find the URL for webpage that presents all the bookies’ odds for a single match. This is done by utilizing the teamnames taken from *upcoming-fixtures.csv*, creating, for each upcoming match, the pattern:

```
"/.*>".$teamNames[0]." - ".$teamNames[1]."<.*/"
```

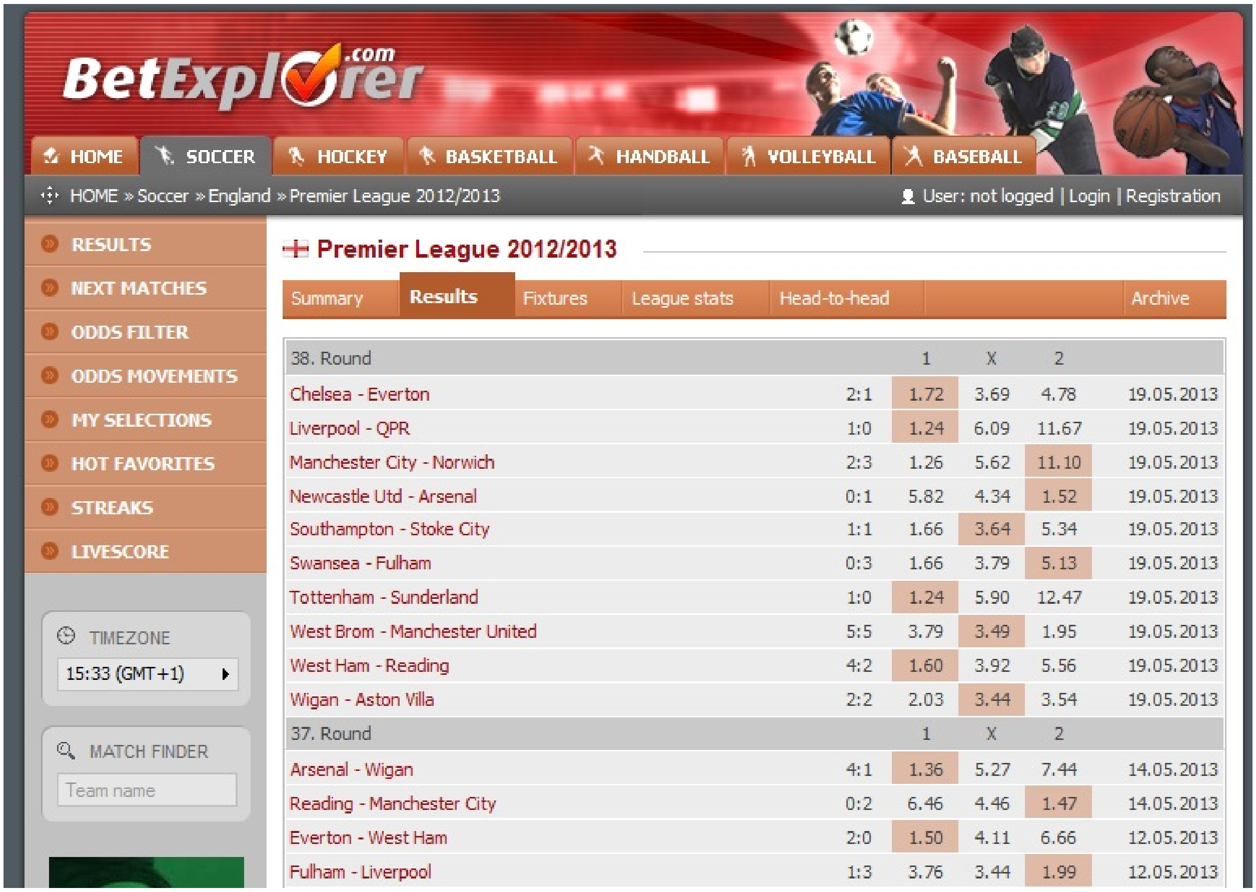
where $teamNames is an array containing the names of the two teams playing eachother. The match we find contains the URL we are after, inside a href-attribute:

```
>Chelsea - Everton</a></td><td class="result"><a
```

```
href="../matchdetails.php?matchid=CSiF8w91" onclick="win(this.href, 500, 500,
0, 1); return false;">2:1</a>
```

Now having the final URL, we can extract the match odds provided by the different bookies. First we identify the *lines* in the code where odds are given, using the pattern:

```
"/(<tr><th.*< \/td>< \/tr>)|(<tr class= \"strong \"><th class= \"first-cell nobr
\">.*< \/td>< \/tr>)/"
```



Premier League 2012/2013

38. Round		1	X	2	
Chelsea - Everton	2:1	1.72	3.69	4.78	19.05.2013
Liverpool - QPR	1:0	1.24	6.09	11.67	19.05.2013
Manchester City - Norwich	2:3	1.26	5.62	11.10	19.05.2013
Newcastle Utd - Arsenal	0:1	5.82	4.34	1.52	19.05.2013
Southampton - Stoke City	1:1	1.66	3.64	5.34	19.05.2013
Swansea - Fulham	0:3	1.66	3.79	5.13	19.05.2013
Tottenham - Sunderland	1:0	1.24	5.90	12.47	19.05.2013
West Brom - Manchester United	5:5	3.79	3.49	1.95	19.05.2013
West Ham - Reading	4:2	1.60	3.92	5.56	19.05.2013
Wigan - Aston Villa	2:2	2.03	3.44	3.54	19.05.2013
37. Round		1	X	2	
Arsenal - Wigan	4:1	1.36	5.27	7.44	14.05.2013
Reading - Manchester City	0:2	6.46	4.46	1.47	14.05.2013
Everton - West Ham	2:0	1.50	4.11	6.66	12.05.2013
Fulham - Liverpool	1:3	3.76	3.44	1.99	12.05.2013

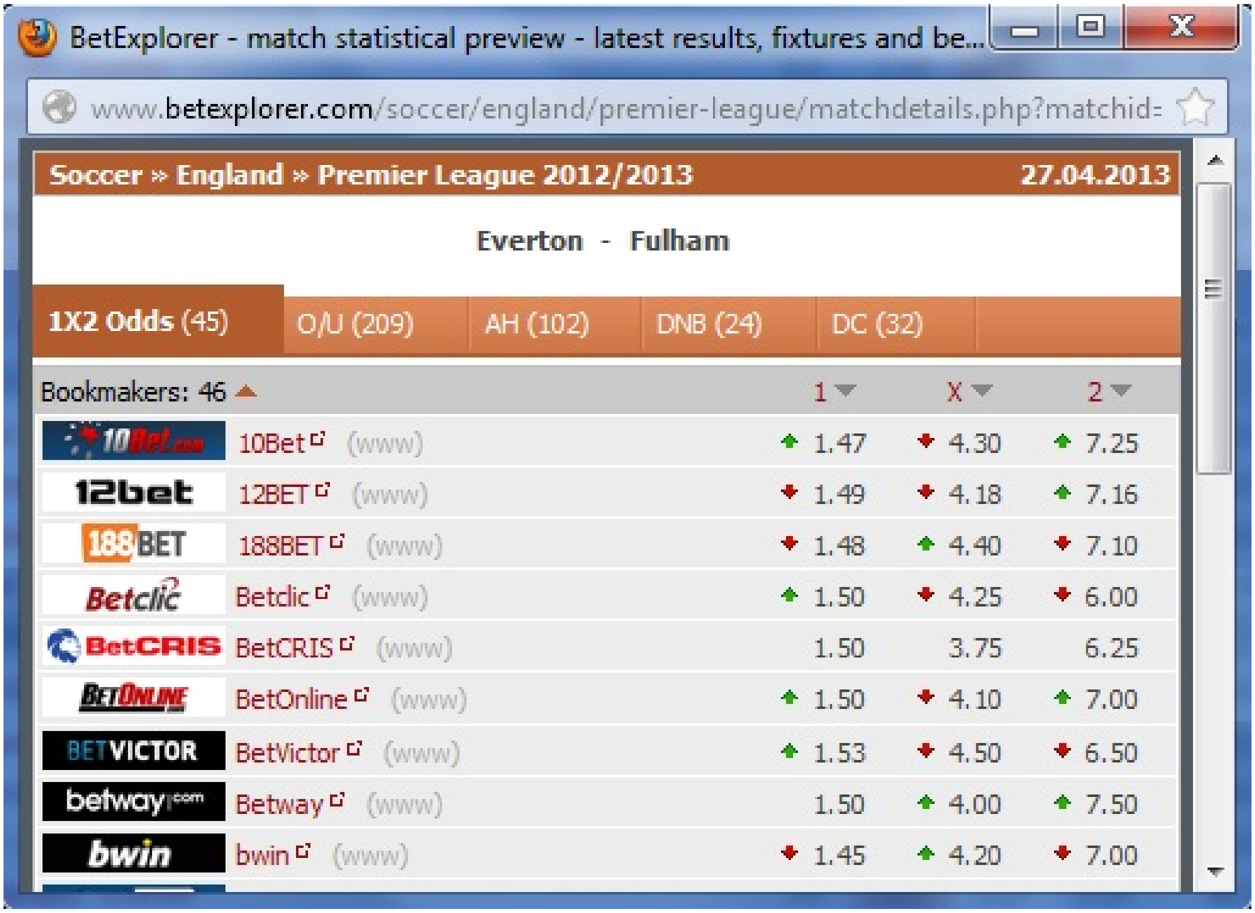
Figure 3.3: Premier League main page presenting average odds for the latest games

When this is done, we locate the odds values and bookie names for each line. For each line, only odds values have the form of a single or double digit, followed by a comma, and then another double digit. The pattern `'/ \" \d \d? \. \d \d \"/'` find all such occurences.

We would like to have the name of each bookmaker which provides the best odds for any of the three possible outcomes, and so for each line, we find bookie names by using the pattern `"/<\/span>.{1,30}<\/a>/"` . When the best bookies are found, these are added to *odds.csv*, which holds odds and bookie info for all games played in the league we are interested in.

Concatenate .csv files. We now have all we need, and can add the odds in *odds.csv* to the fixture info in fixture-info.csv. These may not have the same length, as odds are aquired for both past and potentially future games, whereas match info is of course only gathered for games already played. An example line from the final .csv file, *crawled-PL.csv* could be:

```
614051,13,16,Arsenal,Sunderland,18/08/12,D,10,10,3,0,23,23,12,22,16,2,7,703,637,70,
14,12,0,1,1,2,0,4,4,9,18,28,1,0,294,222,29,12,14,0,1.46,4.75,8.71
```



BetExplorer - match statistical preview - latest results, fixtures and be...

www.betexplorer.com/soccer/england/premier-league/matchdetails.php?matchid=

Soccer » England » Premier League 2012/2013 27.04.2013

Everton - Fulham

Bookmakers: 46 ▲	1	X	2
10Bet (www)	▲ 1.47	▼ 4.30	▲ 7.25
12BET (www)	▼ 1.49	▼ 4.18	▲ 7.16
188BET (www)	▼ 1.48	▲ 4.40	▼ 7.10
Betclic (www)	▲ 1.50	▼ 4.25	▼ 6.00
BetCRIS (www)	1.50	3.75	6.25
BetOnline (www)	▲ 1.50	▼ 4.10	▲ 7.00
BetVictor (www)	▲ 1.53	▼ 4.50	▼ 6.50
Betway (www)	1.50	▲ 4.00	▲ 7.50
bwin (www)	▼ 1.45	▲ 4.20	▼ 7.00

Figure 3.4: Odds for a single match for several bookies, presented by betExplorer.com

The *crawled-PL.csv* file is then used by the simulator to be translated into a .MAT file, which we will come to in the next section.

3.2 The Betting Simulator

The simulator has a quite simple design, with fairly few classes. This section will be going through what each class provides, and how they are tied together. The simulator has been written by our supervisor, Helge Langseth, and unless explicitly stated, the code mentioned has not been altered in any way. Figure 3.5 provides an overview of the architecture, where a blue diamond indicates '*composed of*'.

3.2.1 Game

Game holds information about a game, either an observed one or one with sampled results. it hold some properties such as identifiers for home team and away team as well as result, and also is used as an interface through functions that textually present games.

A game objects properties are:

Property	Description
homeTeamName	the text-version of the team
homeTeamIdx	an index in the range 1 to the amount of teams in the league
awayTeamName	the text-version of the team
awayTeamIdx	an index in the range 1 to the amount of teams in the league
round	an index in the range 1 to the amount of rounds in the league
homeGoals	a positive integer value or distribution over it if it is a simulated result.
awayGoals	a positive integer value or distribution over it if it is a simulated result.
winner	a vector of length 3. Each element relates to the probability of home, draw, or of away victory. If the result is known, the elements will for instance be [0, 1, 0] to signify a draw.
goalDistributionHome	the probability of 0, 1, 2, etc. goals for the home team. This is only valid when used with samples
goalDistributionAway	the probability of 0, 1, 2, etc. goals for the away team. This is only valid when used with samples
gameSimulated	If this is a 1 it is a simulated result, otherwise it is an observation

Table 3.1: Game properties

3.2.2 GameList

A class that is essentially a list of *Game*-objects. Used by Database-instances for organizing matches.

3.2.3 Database

The *Database* class is the interface to the data; any data we wish to retrieve, present or manipulate is done through this class. If the database object currently has actual results stored for games up to and including the 20th round, we can the through the functions in this class hide variables such as goals scored, shots taken etc. in order to simulate an earlier

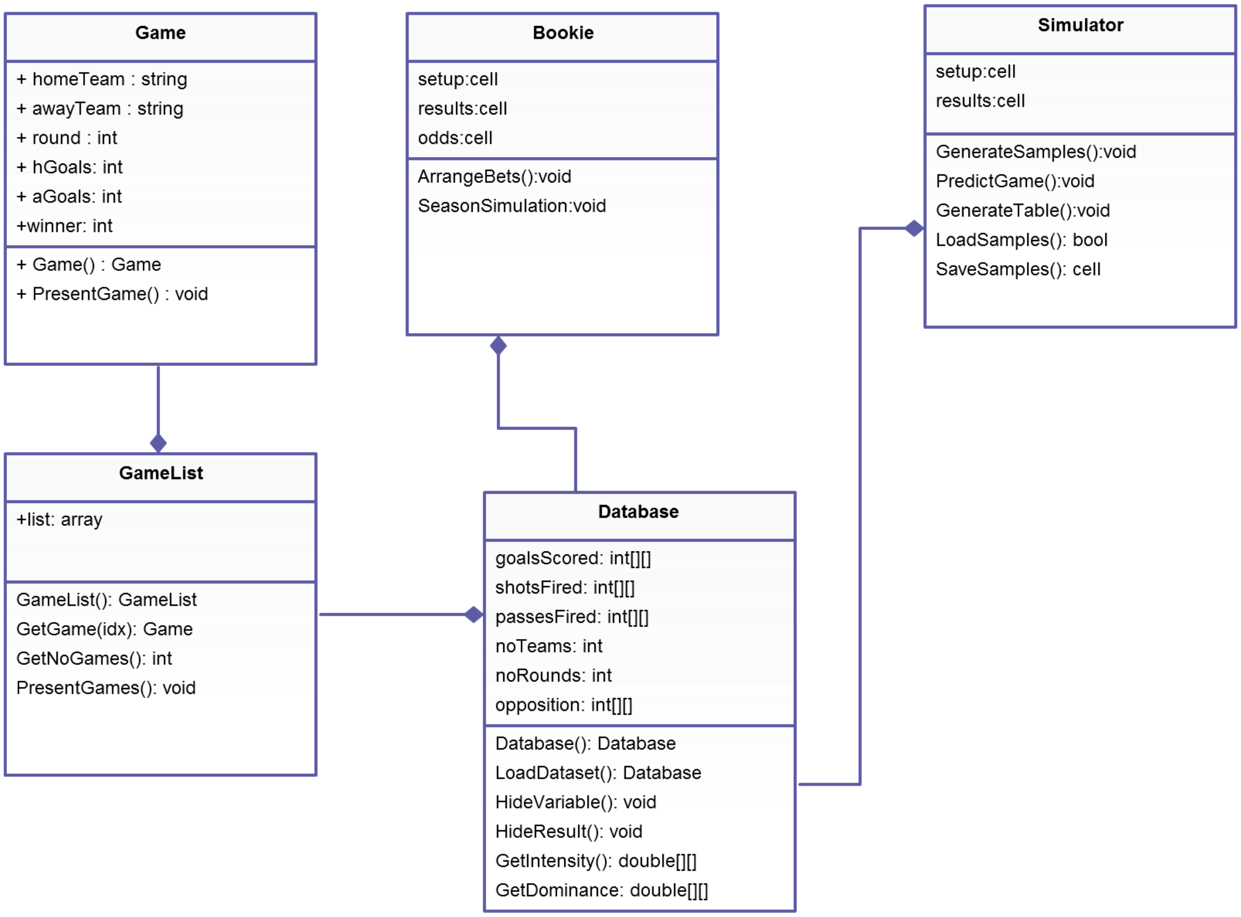


Figure 3.5: Class diagram for the MATLAB betting simulator.

round as if it had not yet been played. This class has been slighty modified to include further match-statistics beyond goals and shots, and functions have been added to hide these variables as well.

A Database objects properties are:

Property	Description
noRounds	number of rounds in a season
teamNames	array of all names of teams in the league
playsAtHome	if playsAtHome (i, t) = 1, then team i plays at home in round t
opposition	if opposition(i, t) = j, then team i plays team j in round t.
gameOrder	if gameOrder(i, t) = j, it means that the t'th game of the season played by team i was the one initially scheduled as round j. Mostly, gameOrder(i, t) = t, but postponing a matches will cause a trickle-down effect.
goalsScored	goalsScored(i,t) gives the amount of goals scored by team$i$ in round t
firedShots	firedShots(i,t) gives the amount of shots fired by team i in round t
shotsOnTarget	shotsOnTarget(i,t) give the amount of shots on target for team i in round t
shotsBlocked	shotsBlocked(i,t) give the number of shots by team i that were blocked, in round t
possession	possession(i,t) gives how large portion of the time team i had the ball in round t
firedPasses	firedPasses(i,t) give attempted passes player for team i in round t
passesOnTarget	passesOnTarget(i,t) gives successful passes for team i in round t
wonContest	wonContest(i,t) covers how many free balls were won for team i in round t
tackles	tackles (i,t) gives the amount of tackles won by team i in round t
airials	airials(i,t) gives the amount of airial duels won by team i in round t
season	string indicating which season we are looking at
league	string indicating which league we are looking at.

Table 3.2: Database properties

3.2.4 Simulator

This class is our interface to JAGS, enabling us to build Markov chains and generate samples. It also contains several plotting functions that present data to us for evaluating models. For generating samples, a Simulator is passed a Database object, which, depending on which round we wish to simulate, may have hidden some of its variables.

The Simulator has two struct properties; `setup` and `results`. `Setup` contains all the variables we need to initiate JAGS, whereas `results` is where values are saved when JAGS has completed. Which values are stored in `results` depend on which values we pass to `setup`, though some variables are set as default should we not pass any arguments at all.

`Results` has two fields, `samples` and `stats`. `Samples` contains all the values JAGS sampled for each variable we initially added to the `monitoring` field of `setup`. `Stats` then contains the mean and standard deviation for each of these variables as well.

The Simulator- function `GenerateSamples()` has been modified to accept new models, as well as include new `Database`-parameters when initiating JAGS. Some functions have also been included for statistical analysis, such as plotting the models effectiveness over the course of the season against each other.

3.2.5 Bookie

This class takes care of testing the different models we build with regards to placing bets and winning money. The main function we will use is its `ArrangeBets()`-function, which takes a Simulator and a Database object as arguments, and prints out the monetary results the given model that produced the Markov Chain produced.

3.2.6 Footy

Is not itself a class, but rather a script that starts the simulator. Bookie, Simulator and Database objects are initialized in this script, and functions that generate samples, places bets, plots diagrams or compares models are run here. This script has been modified to accomodate the new model as well as running new statistical functions in `Simulator`

3.2.7 readData

A script for reading from the *crawled-PL.csv* file and with this data, update the tables in the .mat data files, so that the Simulator and Database objects are using the newest data available for the league. This utilizes an externally written script obtained from `www.stackoverflow.com`[20] which deals with importing csv-files of arbitrary sizes into matrices. `readData` has been modified to add more variables into the MATLAB-matrix, such as possession, tackles, passes and more.

4 Comparing the Models

This chapter summarizes the findings of our preliminary project in Ellefsrød(2012), included for motivating the model developed in chapter 5. We first explain the Maher (1982) and Dixon & Coles (1997) models in greater detail, before presenting the experimental plan, setup and results of the testing done with them. Starting with the oldest and simplest, we first look at the model proposed in the work of Maher (1982), before continuing to that of Dixon & Coles (1997), which largely builds upon the first model

4.1 Maher

Maher (1982) adopts an independent Poisson model for scores, assuming that when two teams i and j are playing each other, and the observed score is (x_{ij}, y_{ji}), X_{ij} is Poisson with mean $k\alpha_i\beta_j$, and that Y_{ji} is Poisson with mean $\alpha_j\beta_i$, and that X_{ij} and Y_{ij} are independent. Here, α_x denotes the attacking strength of team x, β_x, denotes the defending strength of team x, and k denotes the constant factor which indicates the advantage of playing at home. The independence of X_{ij} and Y_{ij} allows the following simplification:

$$\Pr(X_{ij} = x, Y_{ij} = y | k, \alpha_i, \alpha_j, \beta_i, \beta_j) = \text{Poisson}(X = x; k\alpha_i\beta_j)\text{Poisson}(Y = y; \alpha_j\beta_i)$$

Though Maher(1982) use maximum likelihood estimators for α and β, we will use Gibbs sampling to estimate these values for each team. The estimators Maher(1982) use, are:

$$\hat{\alpha}_i = \frac{\sum_{j \neq i}(x_{ij} + y_{ji})}{(1 + \hat{k}^2)\sum_{j \neq i}\hat{\beta}_j}$$

$$\hat{\beta}_j = \frac{\sum_{i \neq j}(x_{ij} + y_{ji})}{(1 + \hat{k}^2)\sum_{i \neq j}\hat{\alpha}_i}$$

$$\hat{k}^2 = \frac{\sum_i \sum_{j \neq i} y_{ij}}{\sum_i \sum_{j \neq i} x_{ij}}$$

Maher(1982) uses the same home advantage factor for each home-team.

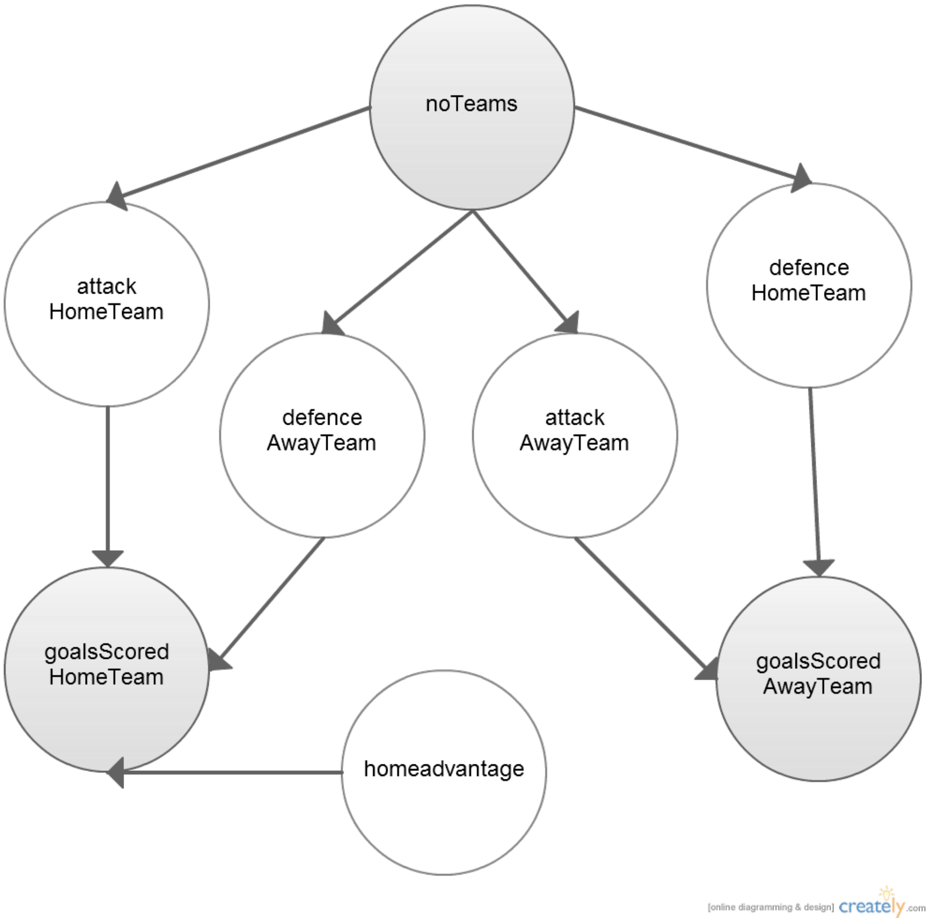


Figure 4.1: The Maher (1982) model constructed as a Baysian network. *Goalsscored (X and Y)* and *noTeams* are the missing values in known data, and *attack* (α), *defence (β)* and *homeadvantage (k)* are model parameters.

4.2 Dixon and Coles

Dixon & Coles (1997) improve on the assumptions made by Maher. They tweak the probability distribution of goals used by Maher (1982) in the following way:

$$\Pr(X_{ij} = x, Y_{ij} = y|\lambda, \mu) = \tau_{\lambda,\mu}(x, y)\text{Poisson}(\lambda)\text{Poisson}(\mu)$$

where $\lambda = \alpha_i\beta_j\gamma$ and $\mu = \alpha_j\beta_i$. In this model, τ determines whether the probability should be adjusted, based on what the scores are. It is computed in the following manner:

$$\tau_{\lambda,\mu}(x, y) = \begin{cases} 1 - \lambda\mu\rho & \text{if } x = y = 0, \\ 1 + \lambda\rho & \text{if } x = 0, y = 1\ , \\ 1 + \mu + \rho & \text{if } x = 1, y = 0, \\ 1 - \rho & \ \text{if } x = y = 1, \\ 1 & \text{otherwise,} \end{cases}$$

where ρ is a parameter used to quantify the inflation effect, given by $\max(\frac{-1}{\lambda}, \frac{-1}{\mu}) \leq \rho \leq \min(\frac{1}{\lambda\mu}, 1)$. In summary, the effect τ has on the Probability distribution is increasing the probabiliy of low-scoring draws (0-0 and 1-1) while decreasing the probabilities of 1-0 and 0-1 results.

Dixon & Coles (1997) also introduce time-varying attacking and defending abilities of each team, so that for the performance at a time point t, historical data far away from t in time, is of less importance then more recent data.

4.3 Experiments and Results

This section explains some of the problems of comparing the models suggested by earlier research done, before stating the experimental plans, setup and results acheived in order to gain empirical evidence of the accuracy of the models proposed by Maher (1982) and Dixon and Coles (1997).

4.3.1 Experimental Plan

Though Maher (1982) produced a method of assigning strengths to teams and calculating the probability of how many goals scored in a match, he does not attempt to use this method to predict results, or provide any empirical evidence of the accuracy of the model he describes. It would therefore be interesting to build that model and use it for betting, using the English Premier League games played in the season of 2011/2012. Similarly, it would be difficult to compare such results to the model proposed by Dixon & Coles (1997), as they have used match results from a different season as their basis for team strengths and prediction. Rue & Salvesen (2000) use a different betting technique when deciding which result to play, and how much to bet, than that used by Dixon & Coles (1997). This again means that comparing the models on the basis of that article alone, is difficult. We therefore define a standard set of data for the models to use for constructing team strengths and weaknesses, and a standard betting strategy used. This should help show how well the models perform compared to each other.

The data set to be used for each of the models will be the English Premier League match results from the season of 2011/2012. The betting strategy will be explained in detail in the following section.

4.3.2 Betting Strategy

The betting strategy we will use, is borrowed from the article written by Rue & Salvesen (2000).

The idea is to maximize the expected profit, while at the same time reducing the variance of the profit to be below some limit. Maximizing is done by assessing which of the three possible outcomes gives the highest profits.

Outcome	Odds from bookmaker	Probabilities by model	Expected Return
Home win	1.8	0.39	0.70
Draw	3.2	0.22	0.72
Away win	4.0	0.39	1.55

Table 4.1: Profits for match outcomes for an arbitrary game

If we were to get fair odds and probabilities on each outcome, the profit, calculated by the formula:

Expected Return(betting on outcome) = probability(outcome) * Odds(outcome)

should provide the value 1 for each possible outcome. This means that placing random bets each week should result in neither gain nor loss of profits as the amount of weeks approach infinity. The issue then becomes finding which outcome provides the best possible expected returns, where a return below 1 can be interpreted as the bookmaker not paying out the full deserved amount, and an expected return above 1 as the bookmaker paying out more than what is deserved, as indicated by the probability values.

As we can see from table 4.1, according to the probabilities the model has provided, only away win gives profits worth placing a bet on. The next question to answer, is how much credit to place on this bet. This decision is answered by the formula:

$$B_i = \min\left(\beta, \frac{S}{2\theta_i(1-o_i)}\right)$$

Where i indicates which match we are betting on, B is the amount bet, β is the current bankroll we have, S is the scaler we have decided to use, which simply scales the bet to a more readable size, θ_i is the odds for the outcome we are betting on for match i, and o_i is the probability of the outcome we are betting on for match i, as calculated by the model used [21].

The min() function ensures that we do not bet more than we have, which is the current bankroll.

Using the example from table 4.1, and setting our bankroll β to be 100 and scaling the amount bet by setting S to be 10, we see that the amount to bet on the outcome of an away win, is:

$$B_i = \min\left(100, \frac{10}{2 * 4.0(1 - 0.39)}\right) = \min\left(100, \frac{10}{2 * 4.0 * 061}\right) = 2.04$$

And thus, a bet of 2.04 credit will be placed on the outcome of an away win. We should also mention that for simplicity, even though there may be several cases in which there is a profit above 1 for a single match, we choose to only place one bet per match, choosing the outcome with highest expected return per credit betted.

4.3.3 Experimental Setup

We build the models by utilizing the JAGS program for model building, and post-processing using the numerical computing environment MATLAB. A MATLAB framework for utilizing the JAGS models were provided by supervisor Helge Langseth, together with an implementation in JAGS of the Maher (1982) model. Interfacing functions between MATLAB and JAGS was written by Mark Steyvers, based on the interface matbugs written by Kevin Murphy and Maryam Mahdaviani [16]. The Gibbs sampler was then called by the MATLAB function `matjags(.)`:

```
[obj.results.samples, obj.results.stats] = matjags( ...
inputData, ...  % Observed data
fullfile(pwd, ['/JAGS/' obj.setup.JAGSfile]), ...  % model def
initials, ...   % Initial values for latent variables
'doparallel' , obj.setup.doparallel, ...  % Parallelization flag
'nchains', obj.setup.nchains,...  % Number of MCMC chains
'nburnin', obj.setup.nburnin,...  % Number of burnin steps
'nsamples', obj.setup.nsamples, ...  % Number of samples to extract
'thin', obj.setup.nthinning, ...  % Thinning parameter
'monitorparams', obj.setup.monitoring, ...  % List of latent variables to monitor
'savejagsoutput' , 1 , ...  % Save command line output produced by JAGS?
'workingDir', fullfile(pwd, '/JAGS/TMP/'), ...
'verbosity' , obj.setup.verbose , ...  % 0:  no text output, 1:  medium, 2:full
'cleanup' , 1 ); % clean up of temporary files?
```

The variables of note here are: *inputData* which contains observed data such as goals scored for each team in every game of the season (depending on where we are in the simulation, some of these may be unobserved and not set), number of teams in the league, number of rounds to play, and the schedule for all the matches; *initials*, which contain the initial values of the latent variables we are going to sample, such as attacking and defending abilities, and the home advantage (conversely, away disadvantage, which is equal and opposite of home advantage). *Verbosity* sets how detailed the output of the Gibbs sampler will be.

The source code of the Dixon & Coles model (1997) can be found in Appendix A.

For each model, we use the following constants for the Gibbs sampler:

Thinning: 30

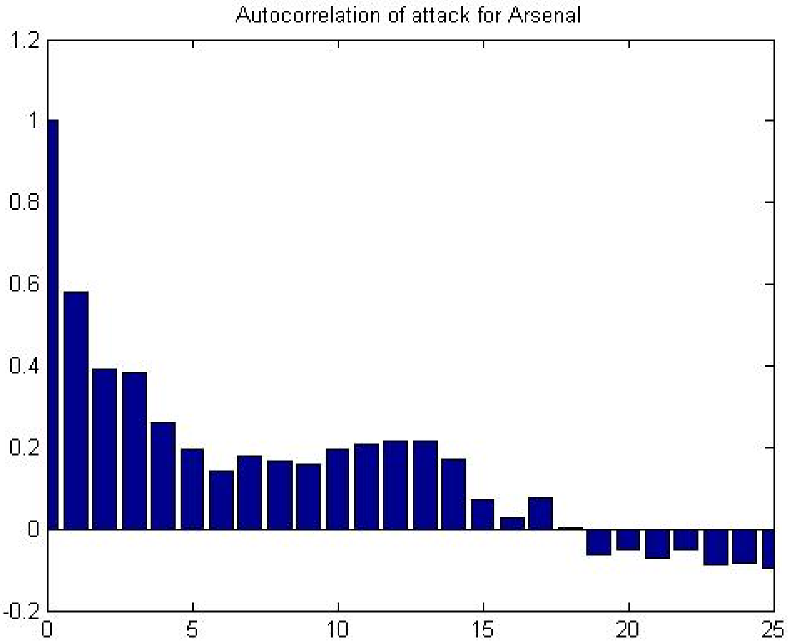


Figure 4.2: Autocorrelation between attack samples for Arsenal, from the Maher(1982) model, with thinning parameter at 1.

The thinning parameter sets how many samples we discard between each we keep. By increasing the thinning parameter, we decrease the autocorrelation between each consecutive sample. Figures 4.2- 4.4 plot the autocorrelation between the samples for the attacking ability of Arsenal, when using the Maher(1982) model. As we can see, deciding not to thin at all, by having the parameter be 1, there is a significant autocorrelation between subsequent samples. This is evident since, by definition, the samples of a Markov chain are dependent of the neighboring sample. We wish to achieve a low autocorrelation between samples, so that there is a larger likelihood of visiting the entire sample space for each variable, thereby constructing a better approximation to the posterior distribution.

Testing again with thinning at 100, shown in figure 4.3, we see that the autocorrelation drops significantly.

A last test shows the correlation at thinning 30 samples. Comparing figures 4.3 and 4.4, there are only slight differences. As the running time of the program is linear with the

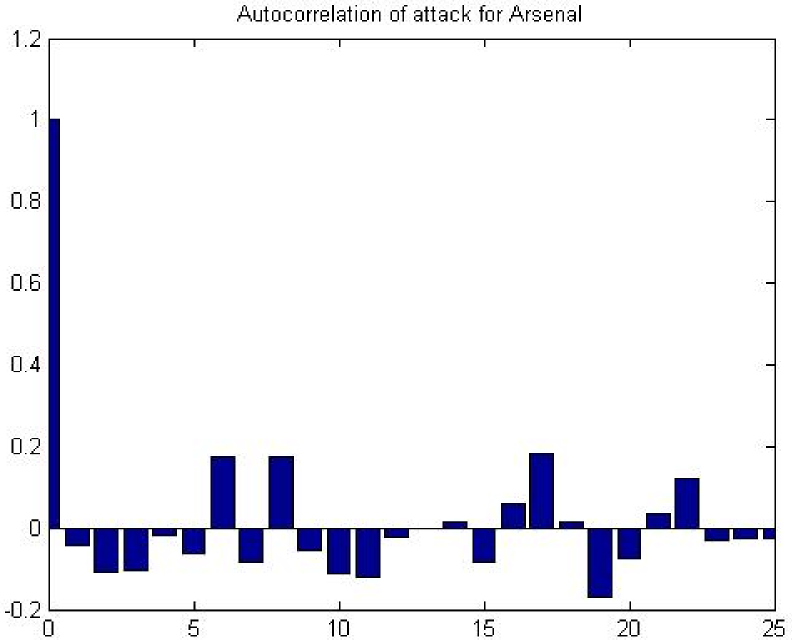


Figure 4.3: Autocorrelation between attack samples for Arsenal, from the Maher(1982) model, with thinning parameter at 100.

size of the thinning parameter, there is a significant trade-off to consider between reducing autocorrelation and keeping the run-time low. Since the reduction in autocorrelation seems to stagnate, we chose to use the value of 30.

Burn-in: 1000

When first starting, the Gibbs sampler will select random sample-values, and as a consequence of random walks it may take a significant amount of time before it reaches the stationary distribution of the Markov chain. We therefore set the burn-in value to 1000, ignoring the first 1000 samples we obtain.
Samples: 10000

Data-set: Match results from each of the 380 matches played in the 2011-2012 season of the English Premier League

For the bookmaker odds for each match, we use the odds presented by the bookmaker WilliamHill. This bookmaker has an average gain per match of 6.26%, meaning that, on average, the sum:

$$TotalOdds = \frac{1}{o_h} + \frac{1}{o_d} + \frac{1}{o_a} = 1.0626$$

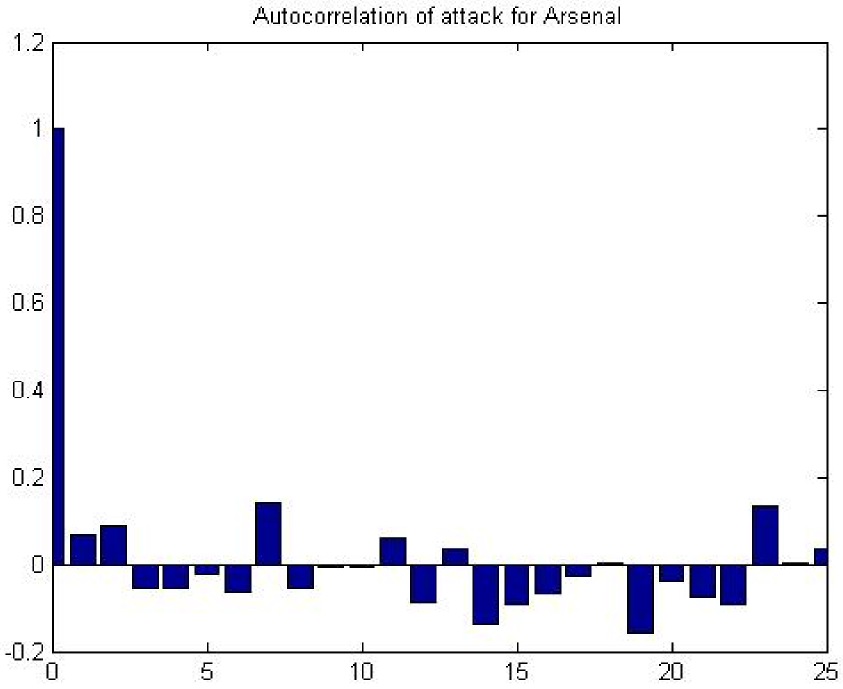


Figure 4.4: Autocorrelation between attack samples for Arsenal, from the Maher(1982) model, with thinning parameter is 30.

is the amount placed on each bet will be scaled by a factor S, explained in section 4.3.2. This scaling factor is set to be 30.

Figure 4.5 shows how the training set increases with each round we play, and the test data is the current round to be played. We will let the Gibbs sampler use observed match-results as the training set S, and the next round (in figure 4.5 this is first round 19, then 20) as the test set T. For each subsequent round, we will model team strengths all over, having now added the previous test set T to the training set S.

We will use round 20 as the first test set, and all preceeding rounds will be the training set. We will thus in total test the predicting accuracy of the model on the last 19 rounds of the season.

4.3.4 Experimental Results

Tables 4.2 - 4.6 provide results for the two models provided by Maher(1982) and Dixon & Coles (1997), respectively. The table-columns present:

1 (Round): Which round is represented

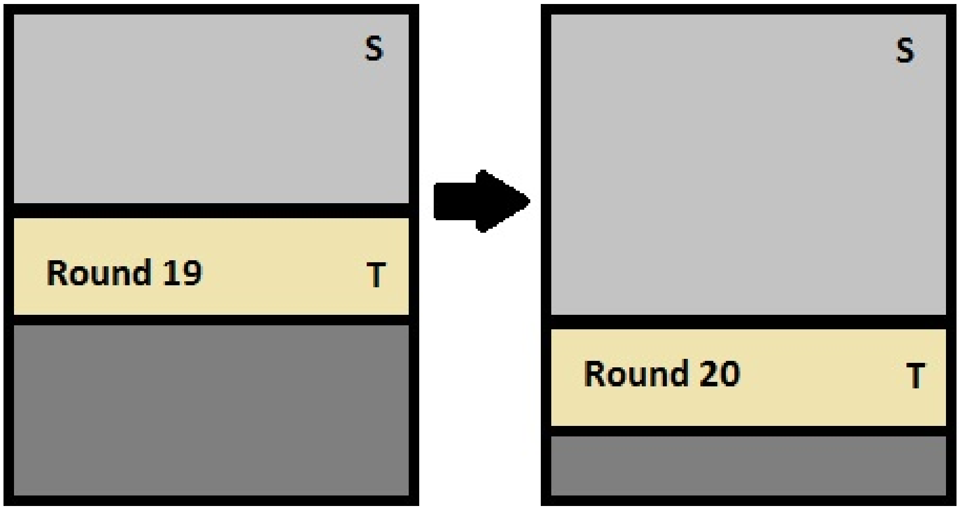


Figure 4.5: A depiction of how the training set S increases with time.

2 (Matches): By comparing our probabilities with the odds given by the bookmaker WilliamHill, we can assess whether it is justifiable to place a bet on a given match, or not. This column shows how many of the 10 matches in each round was considered profitable (expected return higher than 1 for one or more of the outcomes), and subsequently placed bets on.

3 (Bets): This column presents the total amount of credit placed on bets for a given round.

4 (Returns): Presents the total amount of rewards received for the given round

5 (Acc. Bets): Accumulated amount of credit placed on bets up to a point in time.

6 (Acc. Ret.): Accumulated returns, the total amount of credit we have won, including the given round.

7 (Diff): The difference in *accumulated returns* and *accumulated bets.*

8 (Return %): A percentage presenting the winnings for the given round, by the formula:

$$Gains = \frac{Totalprofits - Totalbettings}{Totalbettings}$$

The Maher Model

As we can see from table 4.3, only 2 of the last 10 rounds of betting gave positve payback.

Round	Matches	Bets	Returns	Acc. Bets	Acc. Ret.	Diff	Return %
20	9	73.19	121.16	73.19	121.16	47.97	65.54%
21	10	105.63	175.88	178.82	297.04	118.21	66.5%
22	9	66.27	131.68	245.09	428.72	183.62	98.7%
23	9	120.02	97.59	365.11	526.31	161.20	-18.69%
24	6	51.89	82.88	417.00	609.19	192.19	59.72%
25	9	101.67	127.66	518.67	736.85	218.17	25.56%
26	10	86.39	70.23	605.06	807.08	202.02	-18.70%
27	10	182.94	221.23	788.00	1028.31	240.31	20.93%
28	8	71	100.38	859.00	1128.69	269.06	40.14%

Table 4.2: Results of betting with the model proposed by Maher (1982), rounds 20-28

Round	Matches	Bets	Returns	Acc. Bets	Acc. Ret.	Diff	Return %
29	9	118.92	91.11	977.92	1219.8	241.24	-23.39%
30	10	90.30	99.95	1068.22	1319.75	250.89	10.69%
31	10	89.03	71.99	1157.25	1391.74	233.84	-19.15%
32	8	51.60	47.93	1208.85	1439.67	230.17	-7.12%
33	10	107.75	48.65	1316.6	1488.32	171.07	-54.85%
34	9	78.74	145.44	1395.34	1633.76	237.77	84.70%
35	8	65.20	50.52	1460.54	1684.28	223.09	-22.52%
36	9	108.52	87.95	1569.06	1772.23	202.51	-18.96%
37	10	71.27	39.37	1640.33	1811.6	170.62	-44.76%
38	10	46.73	0	1687.06	1811.6	123.89	-100%

Table 4.3: Results of betting with the model proposed by Maher (1982), rounds 29-38

However, the bets placed in table 4.2 were highly successful, earning credit in 7 out of 10 rounds. We may also notice that when winning, we generally had a profit margin above 50%, while when loosing, it was more commonly at a low rate of 20% of bets placed.

Overall, we gained 124 credit when betting a total amount of 1687, a 7.4 % return, which would indicate that the model suggested by Maher(1982) was quite successful.

There are several possible explanations to why the model does so poorly the last 10 rounds:

- The model proposed by Maher (1982) does not have dynamic attacking and defending abilities per team. Instead, they are represented by a single value, and do not vary over time. A good example of a team which developed over the season is Wigan, which started the season very badly, but during the latter stages of the season picked up some momentum and managed to climb to a safe spot on the table, finishing in a 15th place, as shown in Figures 1 and 2 in Appendix A.1.

  With the Maher model, in the 34th round, each match result from the whole season up to that point will have an equal say when calculating the outcome. Thus the recent good form of teams such as Wigan is not captured, and the probabilities of Wigan's opposition will be artificially high, inducing a bet against Wigan. The table below shows how the model placed bets in the last 8 Wigan matches:

Round	Home team	Away team	Result	Our bet	Winning Bet
31	Wigan	Stoke	2-0	A	No
32	Chelsea	Wigan	2-1	A	No
33	Wigan	Man Utd	1-0	A	No
34	Arsenal	Wigan	1-2	A	Yes
35	Fulham	Wigan	2-1	H	Yes
36	Wigan	Newcastle	4-0	A	No
37	Blackburn	Wigan	0-1	H	No
38	Wigan	Wolverhampton	3-2	A	No

Table 4.4: Shows how the Maher(1982)-model fared when trying to anticipate the last Wigan matches of the season.

As we can see, 5 of the 8 last matches Wigan played, the model failed to recognize that Wigan was in a good form. In total, it only got two of these matches correct.

- The model does not recognize that when the season is coming to an end, there are certain teams fighting to avoid relegation, and certain other teams fighting for the premiership, or qualifying for the Champions League, or more still fighting to qualify for the Europa League (this last factor may have less of an impact, as it seems most of the high profile Premiership teams do not place this tournament in high regards). This possibility to win the league or avoid relegation may have a significant impact on the mentality of two opposing teams at closing matches of the season.

- As we have mentioned previously, several studies have indicated that the unaltered Poisson distribution that Maher (1982) utilizes for amount of goals scored may not be entirely correct, and that a slightly modified version would be better suited. This is the case of the Dixon & Coles (1997) model, which we shall evaluate next.

The Dixon & Coles Model

Similarly to the preceeding model, this one also proves efficient during the 3rd quarter of the season, profitting on 7 of the 9 rounds presented in table 4.5.

The differences between the two models are more apparent in table 4.6, where this model has an overall increase in earnings from round 29 to 38, 36 credit, and profitting on exactly half of the rounds. Comparably, the preceeding model went from 269 credit earned in round

Round	Matches	Bets	Returns	Acc. Bets	Acc. Ret.	Diff	Return %
20	10	62.46	115.20	62.46	115.20	52.74	84.44%
21	10	46.12	73.52	108.58	188.72	80.15	59.43%
22	10	59.09	94.79	167.67	283.51	115.85	60.41%
23	10	63.11	54.63	230.78	338.14	107.36	-13.44%
24	8	62.57	106.02	293.35	444.16	150.81	69.44%
25	10	61.98	93.04	355.33	537.2	181.88	50.12%
26	9	48.66	0	403.99	537.2	133.22	-100%
27	9	56.17	49.54	460.16	586.74	126.59	-11.81%
28	10	57.58	105.45	517.74	692.19	174.46	83.14%

Table 4.5: Results of betting with the model proposed by Dixon & Coles (1997), rounds 20-28

Round	Matches	Bets	Returns	Acc. Bets	Acc. Ret.	Diff	Return %
29	10	82.24	80.75	599.98	772.94	172.97	-1.82%
30	9	42.21	65.12	642.19	838.06	195.87	54.26%
31	10	62.10	49.10	704.29	887.16	182.88	-20.93%
32	10	60.27	48.49	764.56	935.65	171.09	-19.55%
33	10	115.72	174.94	880.28	1110.59	230.31	51.18%
34	9	37.66	61.32	917.94	1171.91	253.97	62.81%
35	8	54.51	66.18	972.45	1238.09	265.64	21.42%
36	10	55.42	26.60	1027.87	1264.69	236.82	-52.01%
37	10	79.19	24.73	1107.06	1289.42	182.36	-68.78%
38	10	77.03	99.13	1184.09	1388.55	204.46	28.69%

Table 4.6: Results of betting with the model proposed by Dixon & Coles (1997), rounds 29-38

29 to 124 at the end of the season, a loss of 145 credit. The Dixon & Coles (1997) model has a 17.3 % positive return overall, considerably better at placing bets then the first model tested.This may be accountable by the Dixon & Coles (1997) model being able to provide dynamic strengths to each team, therefore maintaining a more stable assessment of probabilities.

Dixon & Coles (1997) model assesses goal scored by a home-team as:

$$\text{GoalScored}_h = \text{Poisson}(\lambda_h), \lambda_h = \text{attack}_h * \text{defence}_a$$

Where h and a stand for hometeam and awayteam, respectively. This means that increasing the defense of the opposition, increases the expected amount of goals scored by the home team, λ. Thus, better teams have high attack values, and low defence values, whereas lower-ranking teams have high defence values, and low attacking values.

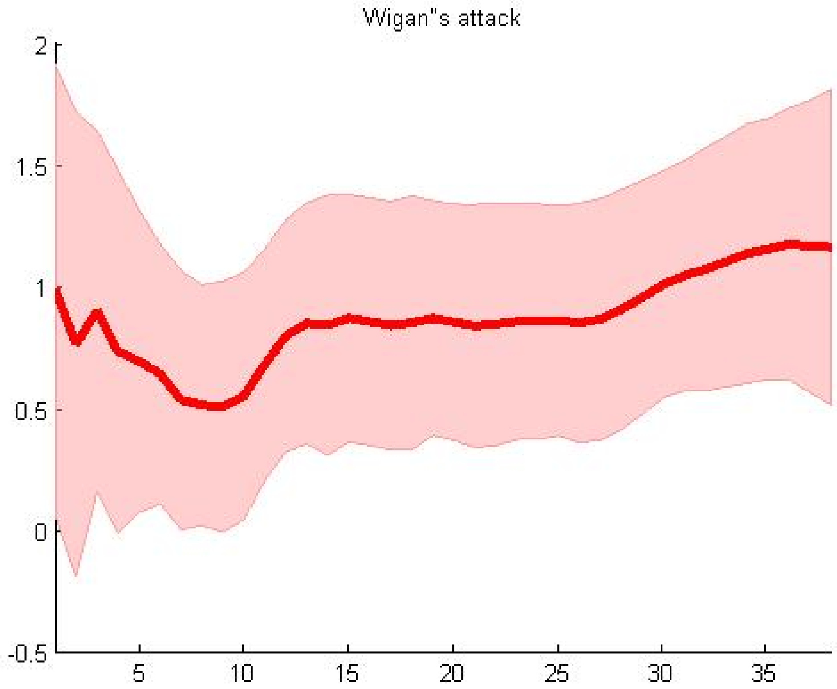


Figure 4.6: Representation of the attacking ability of Wigan Athletic over the course of the season

Figures 4.6 - 4.9 shows how the model assigned attacking and defending abilities to the two teams Wigan Athletic and Manchester City, and how these values varied over the course of the season. The thick line indicates the team's attack (and defence) for each round, averaged over the samples. The area greyed out above and below the line indicates values within double standard deviation distance. As we can see, Wigans attacking ability starts out at a medium level before falling drastically during the first 10 games. It then has a return to a mediocre value before slightly increasing towards the end of the season. Meanwhile, Figure 4.7 shows how the defending ability is severely high during the first half of the season, improving towards the end of the season, enjoing a peak during the final few matches of the season. Looking at figures 1 and 2 of the Appendix A.1, we see how this coincides very well with how Wigan fared during the season, especially the dive to last place during the first 14 weeks, and their improvement over the final 14 weeks.

Looking at Manchester City next, this team provides an entirely different story, placed either first or second during the entire course of the season. This is well represented by the defending ability of the team, staying mostly constant through the season, with the worst values at the first couple of weeks of the season. The attacking abiliy of the team is similar, with a slightly more pronounced peak at the start of the season, during the first 10-12 weeks. This is not surprising, as the team during this period scored 33 goals in their first 9 matches, averaging at a staggering 3.67 goals per match.

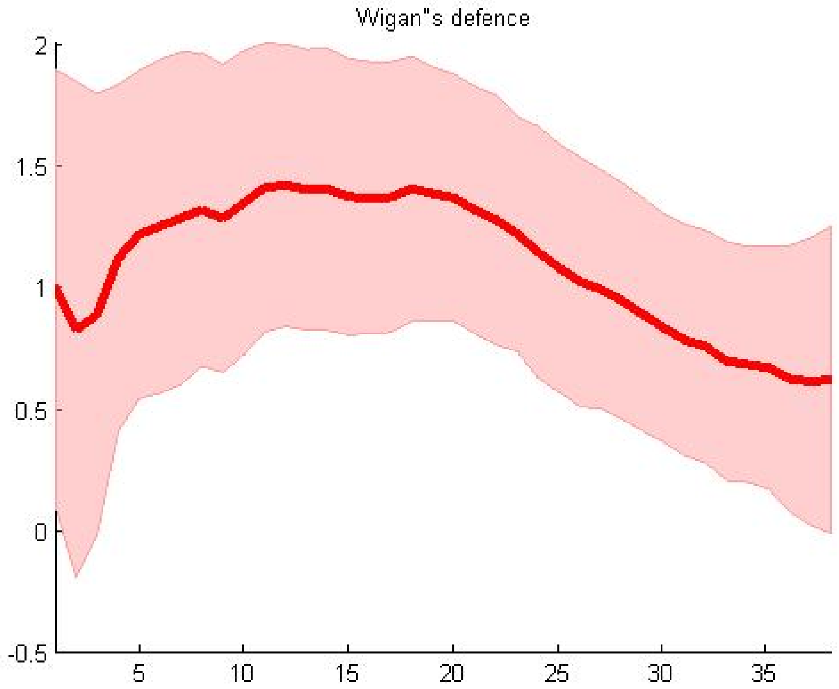


Figure 4.7: Representation of the defending ability of Wigan Athletic over the course of the season

If we again look at the final matches of Wigan and the bets placed by this model, we see from table 4.7 that this model performs better than that of Maher (1982). As we can see, it places a bet on Wigan to win against Manchester United, Arsenal and Chelsea, predicting correct in 2 out of these 3 cases. We must still have in mind that it is not simply a matter of predicting the winner the most often, but getting a positve return on investment from it. Though this model placed a bet on Wigan winning several of the matches presented in Table 4.7, it does not necessarily mean that this was what the model deemed most likely. Rather, comparing it to the odds given by the bookmaker, this was the outcome that would give a highest expected return.

4.4 Evaluation

We found in the previous section that the model proposed by Dixon & Coles (1997) outperform that of Maher (1982), the former earning more than 17% of the amount placed on bets, while the latter managed a 7% increase in bankroll. We must still keep in mind that as Gibbs sampling is a random process that will provide us with non-deterministic results given the same input, for another run the results may therefore have been different. We have tried to make up for this by using a high amount of samples and burn-in, and reducing the

Round	Home team	Away team	Result	Our bet	Accuracy
31	Wigan	Stoke	2-0	H	1
32	Chelsea	Wigan	2-1	A	0
33	Wigan	Man Utd	1-0	H	1
34	Arsenal	Wigan	1-2	A	1
35	Fulham	Wigan	2-1	N/A	N/A
36	Wigan	Newcastle	4-0	H	1
37	Blackburn	Wigan	0-1	A	1
38	Wigan	Wolverhampton	3-2	H	1

Table 4.7: Shows how the Dixon & Coles(1997)-model fared when trying to anticipate the last Wigan matches of the season.

autocorrelation between the samples by the means of thinning.

We have only judged the two models on their moneymaking efficiency, and not tested which of the two proved the most accurate predictions, with regards to the Rank Probability Score presented by Constantinou & Fenton (2012). It could be interesting to do this assessment as well, but for our purpose, the main goal is to find the model which produces the highest income. This may not be the model which provides the best predictions.

Based on the results from the experiment, we will regard the model proposed by Dixon & Coles (1997) as the model which should be attempted to be improved upon.

It should also be clarified that the internet crawler described in chapter 3 was not used for the tests of these models, as they do not utilize the more in-depth statistics found with the crawler. We also already had data for amount of goals scored by each team for this season, and the odds provided by the bookmaker William-Hill.

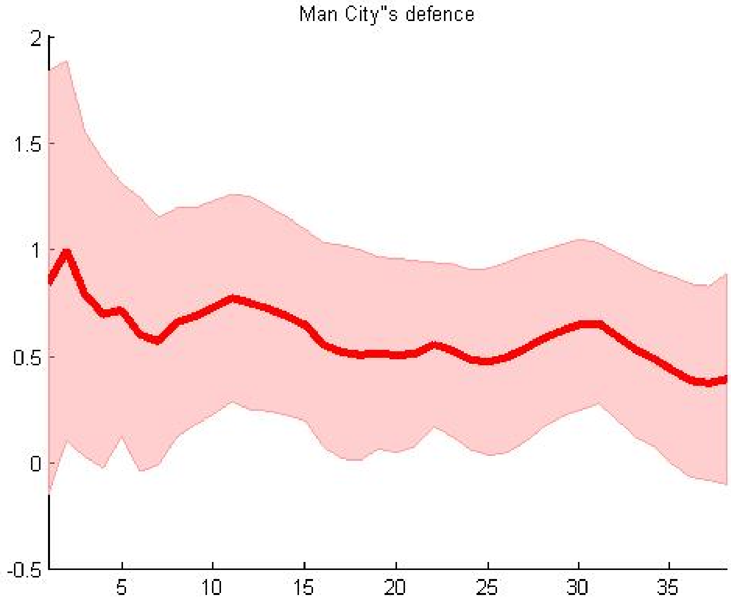


Figure 4.8: Representation of the defending ability of Manchester City over the course of the season.

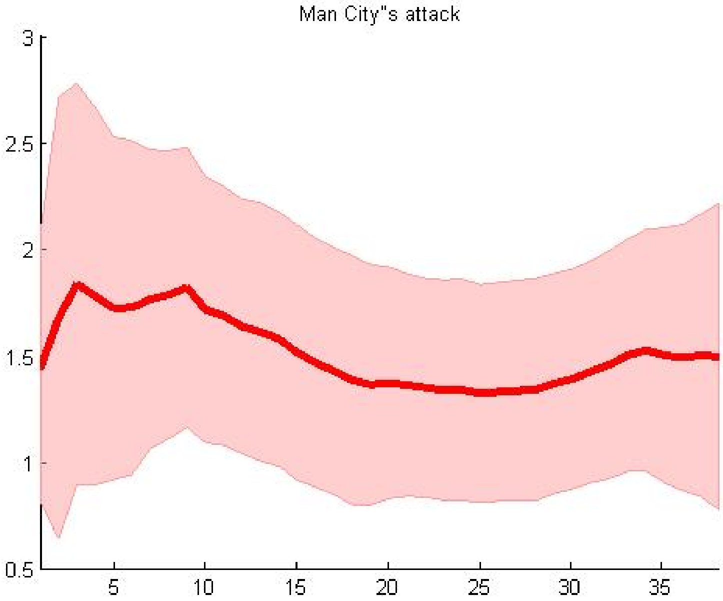


Figure 4.9: Representation of the attacking ability of Manchester City over the course of the season

5 Constructing a Model

This chapter describes the model extensions we have attempted to implement, and why we made the modelling choices we did. We will look at some of the variables that we feel are deemed best at determining a teams defending and attacking skills, and do several tests to see if our hypotheses are correct, before deciding on two sets of variables to be used in our model.

5.1 The statistics available

From the internet crawler, we have obtained the following 17 variables for each team, for each match played in the Premier League season 2012/13:

Property	Description
'home_blocked_scoring_att'	Total shots by home-team that were blocked by away-team players
'home_shot_off_target'	Total shots by team that were off target (includes post-shots)
'home_ontarget_scoring_att'	shots by home-team that were on target
'home_post_scoring_att	Total shots made by home-team that hit the post
'home_total_scoring_att	Total shots made by home-team
'home_won_contest	Total free balls won by home-team
'home_fk_foul_lost'	Amount of fouls commited by home-team
'home_total_throws'	Amount of throw-ins by home-team
'home_total_tackle'	Amount of tackles peformed by home-team
'home_total_offside'	Amount of offsides the home-team were caught in
'home_won_corners'	Amount of corner-kicks won by home-team
'home_total_pass'	Amount of passes attempted by home-team
'home_accurate_pass'	Amount of successful passes by home-team
'home_possession_percentage	How large part of the match the home-team were in possession of ball
'home_aerial_lost	Amount of headers lost by home-team
'home_aerial_won	Amount of headers won by home-team
home_goals	Amount of goals scored by home-team

Table 5.1: Statistical variables obtained with internet crawler. Only home-team statistcs are shown.

Some of the variables from Table 5.1 are reduntant, as mutual exclusiveness means they do not add information. For instance, knowing the total amount of shots made and the amount of shots that were on target, we can deduce how many shots were off target. Likewise when knowing the amount of airial duels won by both team, we also implicitly know the amount they lost. Therefore there are only 15 variables for each team.

We also decide to not use amount of shots that hit the post, because these occur very rarely and often have a zero value. It is also concidered difficult to address the impact of throw-ins and offsides, and so these are also discarded.

5.2 Using the coefficient of determination

The coefficient of determination, R^2, is used in statistical analysis for determining the error of a regression line, compared to using the simple average. The better the linear regression fits the data, the closer the value of R^2 is to one. If the coefficient of determination is exactly as good (or bad) as the simple average, it has the value zero. The mathematical formula for the coefficient is:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

where SS_{res} is the *sum of squares of residuals*, i.e. the sum of the errors of each sample point when using the regression line, and SS_{tot} is the *total sum of squares*, i.e. the sum of errors of each sample point when using the simple average.

The coefficient of determination can be used for evaluating models used for predicting future outcomes on the basis of other information. We will use it by evaluating the variables listed in the previous section, against goals scored in the same or following round. This is to assess how much a factor each variable seems to play when a team scores goals, and how a value from one round passes on to the next. Does a high amount of tackles lead to higher amount of goals? Do few tackes in one round lead to few tackles in the next, and implicitly goals as well?

5.2.1 The predictive nature of goals

The table below shows the R^2 values of goals in rounds t mapped to goals in round *t+1* for each team in the league, and the average value for the league.

As we can see from table 5.2, given that we know how many goals were scored in round t by each team, predicting how many goals they will score in the subsequent round is difficult.

Team	R^2
Arsenal	0.0123
Aston Villa	0.0024
Chelsea	0.0005
Everton	0.0142
Fulham	0.0001
Liverpool	0.0294
Manchester City	0.0218
Manchester United	0.0008
Newcastle	0.0001
Norwich	0.0088
QPR	0.0921
Reading	0.0479
Southampton	0.0045
Stoke	0.0341
Sunderland	0.0021
Swansea	0.0002
Tottenham	0.0012
West Bromwich	0.0026
West Ham	0.1303
Wigan	0.0796
Average	0.0243

Table 5.2: R^2 values for each team for goals scored in round t opposed to round t+1 for t={1,...37}

5.2.2 Using other variables

The table 5.3 shows the R^2 value for the most promising variables taken from table 5.1. As we can see, several of them have a higher R^2 score than that of goals scored in the previous round. The most prominent being the possesional statistics; total attempted passes, total completed passes and possession give significantly higher values than goals scored.

At the bottom of table 5.3 are two additional variables; *Intensity* and *Dominance.* These are defined as follows:

$$I_{basic}(i,t) = \frac{TAC(i,t) + HED(i,t) + CON(i,t)}{TAC(i,t) + HED(i,t) + CON(i,t) + TAC(opp,t) + HED(opp,t) + CON(opp,t)}$$

$$I_{balanced}(i,t) = I_{basic} * POSS(i,t)$$

Variable	R^2 averaged over all teams
Shots on target	0.0209
Total shots fired	0.0145
Shots blocked	0.0160
Total passes	0.0488
Completed passes	0.0440
Possession	0.0459
Headers won	0.0233
Contests won	0.0315
Intensity	0.0508
Dominance	0.0353

Table 5.3: R^2 values for several variables in round t opposed to goals in round t+1

Variable	R^2 averaged over all teams
Shots on target	0.2896
Total shots fired	0.0927
Shots blocked	0.0199
Total passes	0.0170
Completed passes	0.0237
Possession	0.0215
Headers won	0.0479
Contests won	0.0124
Intensity	0.0208
Dominance	0.1231

Table 5.4: R^2 values for several variables in round t opposed to goals in *that round*

$$Intensity(i,t) = \frac{I_{balanced}(i,t)}{I_{balanced}(i,t) * I_{balanced}(opp,t)}$$

where i is a given team, t a given round, *TAC(i,t)* is the amount of tackles, *HED(i,t)* the amount of airial duels won, *CON(i,t)* the amount of contests over free balls won, *opp* the opposition team, and *POSS(i,t)* is the possessional stats.

This statistic tries to capture the intensity of a team, compared to that of its opponent at that time. We use tackling, airial prowess and the willingness to fight for free balls in order to assess the flair of a team. This provides the thought behind I_{basic}. However, scenarios occur where one team sees more of the ball than the other during the course of a match; less time for the opposition possessing the ball will reduce the amount of opportunities for a team to set in a tackle, and likewise increase that same amount for the opposition. $I_{balanced}$ counters this by multiplying the intensity with each teams respective possession statistic. A team

Variable	R^2 averaged over all teams
Shots on target	0.1754
Total shots fired	0.0958
Shots blocked	0.0303
Total passes	0.0280
Completed passes	0.0370
Possession	0.0244
Headers won	0.0591
Contests won	0.0301
Intensity	0.0358
Dominance	0.2442

Table 5.5: R^2 values for several variables in round t opposed to *goal difference* in that round

which has had the ball *more*, will have their Intensity reduced *less* than their opposition. The last equation simply normalizes the $Intensity_{balanced}$.

Dominance is a simpler formula [11]:

$$Dominance(i,t) = \frac{shotsOnTarget(i,t)}{shotsOnTarget(i,t) + shotsOnTarget(opposition,t)}$$

Dominance then depicts how many shots a team got on target compared to its opposition.

5.2.3 Choosing variables

From the R^2 values of table 5.3, *Intensity* gives what seems to be the best predictive results. However, tables 5.4 and 5.5 show that although intensity may be used to predict goals in subsequent matches, it is not the best indicator of goals in the current round. *Shots on target* and *Dominance* seem to be better estimators for assessing goals scored in the current round.

It may be argued that since the R^2 values are so low (most of the variables are only slightly better than the simple average at predicting goals in subsequent matches) , that they can be disregarded as not relevant. We could set up a model using all the variables in tables 5.3 - 5.5, but by using *intensity* and *dominance*, which are constructed by using several of the other variables, we can use a simplified model where more of the variables come into play. As seen by tables 5.6- 5.8, Both variables are only slighty worse than *shots on target*, which is considered a strong indicator of teams' comparable strengths [11]. Since *intensity* encompasses both possession, headers won, contests won and tackles, these will not be used. *Dominance* gives higher values than all three shot statistics, and will also be prioritized when building our model.

Team	Average Intensity	IR	FLP(points)	Diff.
Manchester United	0.5769	3	1 (89)	2
Manchester City	0.5871	2	2 (78)	0
Chelsea	0.5348	8	3 (75)	5
Arsenal	0.6053	1	4 (73)	3
Tottenham	0.5209	9	5 (72)	4
Everton	0.5368	7	6 (63)	1
Liverpool	0.5759	4	7 (61)	3
West Brom	0.4648	15	8 (49)	7
Swansea	0.5170	10	9 (46)	1
West Ham	0.4487	17	10(46)	7
Norwich	0.4145	18	11(44)	7
Fulham	0.4873	11	12(43)	1
Stoke	0.4727	14	13(42)	1
Southampton	0.5388	5	14(41)	9
Aston Villa	0.4739	13	15(41)	2
Newcastle	0.4780	12	16(41)	4
Sunderland	0.4143	19	17(39)	2
Wigan	0.5387	6	18(36)	12
Reading	0.3569	20	19(28)	1
QPR	0.4565	16	20(25)	4

Table 5.6: The average intensity of each team over the entire season. The third column gives the intensity ranking (IR) of each team, the fourth shows the final league position (FLP) of each team with the total points tally in parenthesis, and the final column gives the absolute positional differene between the IR and FLP. The average error in position is 3.8, and the standard deviation of the average intensity is 0.0647.

5.3 The Model

The amount of shots a team is able to get on target during a match compared to the opposition can arguably be determined by four factors:

- The teams attack strength: The better a team is at creating opportunities the more shots it will probably get compared to the opposition.
- The opposing teams attack strength: The better the opposition is at creating opportunities, the more shots they will get. This will reduce the shot dominance the first team has.
- The teams defence strength: The stronger a team is at defending, the less opportunities the opposition will get, increasing the dominance of the team.
- The opposing teams defence: The stronger the opposition is at stopping the first team,

Team	Average Dominance	DR	FLP(points)	Diff.
Manchester United	0.6211	3	1 (89)	2
Manchester City	0.6747	1	2 (78)	1
Chelsea	0.5667	7	3 (75)	4
Arsenal	0.5815	5	4 (73)	1
Tottenham	0.6475	2	5 (72)	3
Everton	0.5722	6	6 (63)	0
Liverpool	0.6204	4	7 (61)	3
West Brom	0.4867	10	8 (49)	2
Swansea	0.4451	13	9 (46)	4
West Ham	0.4277	15	10(46)	5
Norwich	0.4020	18	11(44)	7
Fulham	0.4046	17	12(43)	5
Stoke	0.4313	14	13(42)	1
Southampton	0.5235	8	14(41)	6
Aston Villa	0.4699	12	15(41)	3
Newcastle	0.4812	11	16(41)	5
Sunderland	0.3777	19	17(39)	2
Wigan	0.5029	9	18(36)	9
Reading	0.3455	20	19(28)	1
QPR	0.4179	16	20(25)	4

Table 5.7: The average dominance of each team over the entire season. The third column gives the dominance ranking (DR) of each team, the fourth shows the final league position (FLP) of each team with the total points tally in parenthesis, and the final column gives the absolute positional differene between the DR and FLP. The average error in position is 3.4, and the standard deviation of the average dominance is 0.0965.

the fewer opportunities will be created, reducing their dominance.

A high defence value for a team indicates many goals conceded. Following this, we set up the following formula for dominance:

$$Dominance(i,t) = (attack(i,t) - defence(i,t)) - (attack(opposition,t) - defence(opposition,t))$$

Similarly, the same can be said of intensity:

- The teams attack strength: The better a team is in attack, the more control they have of the ball. This will arguably make it difficult to dispossess them, reducing the amount of tackles and free contests won.

Team	Average Shots on target	SR	FLP(points)	Diff.
Manchester United	5.6053	4	1 (89)	3
Manchester City	6.0526	3	2 (78)	1
Chelsea	5.5526	5	3 (75)	2
Arsenal	4.5789	7	4 (73)	3
Tottenham	6.3421	1	5 (72)	4
Everton	5.4211	6	6 (63)	0
Liverpool	6.1053	2	7 (61)	5
West Brom	4.4737	10	8 (49)	2
Swansea	4.3947	11	9 (46)	2
West Ham	4.1579	14	10(46)	4
Norwich	3.5263	18	11(44)	7
Fulham	4.2105	13	12(43)	1
Stoke	3.0000	20	13(42)	7
Southampton	4.3947	12	14(41)	2
Aston Villa	3.7105	16	15(41)	1
Newcastle	4.5789	8	16(41)	8
Sunderland	3.6579	17	17(39)	0
Wigan	4.4737	9	18(36)	9
Reading	3.3684	19	19(28)	0
QPR	3.8421	15	20(25)	5

Table 5.8: The average amount of shots on target of each team over the entire season. The third column gives the shots on target ranking (SR) of each team, the fourth shows the final league position (FLP) of each team with the total points tally in parenthesis, and the final column gives the absolute positional differene between the SR and FLP. The average error in position is 3.3, and the standard deviation of the average intensity is 0.1072 (after scaling the values to have a mean of 0.5, as intensity and dominance does).

- The opposing teams attack strength: This would infer the opposite results as the previous point. This would lead to the team managing fewer tackles, intercepts and winning headers.
- The teams defence strength: the defence of a team is generally evaluated at how good they are at breaking up play; this can only be done by tackling, heading and intercepting free balls. Having a strong defence will increase *Intensity*.
- The opposing teams defence: Similarly, this will lead to the opposition getting in more tackles, etc., and give the opposition increased *Intensity*.

This gives a similiar formula for intensity:

$$Intensity(i,t) = \omega*(attack(i,t) - defence(i,t)) - (attack(opposition,t) - defence(opposition,t)) \, ,$$

where ω is the weight of *Intensity* compared to *Dominance* when determining attacking and defending abilities. Figure 5.1 depicts the model as a Markov chain.

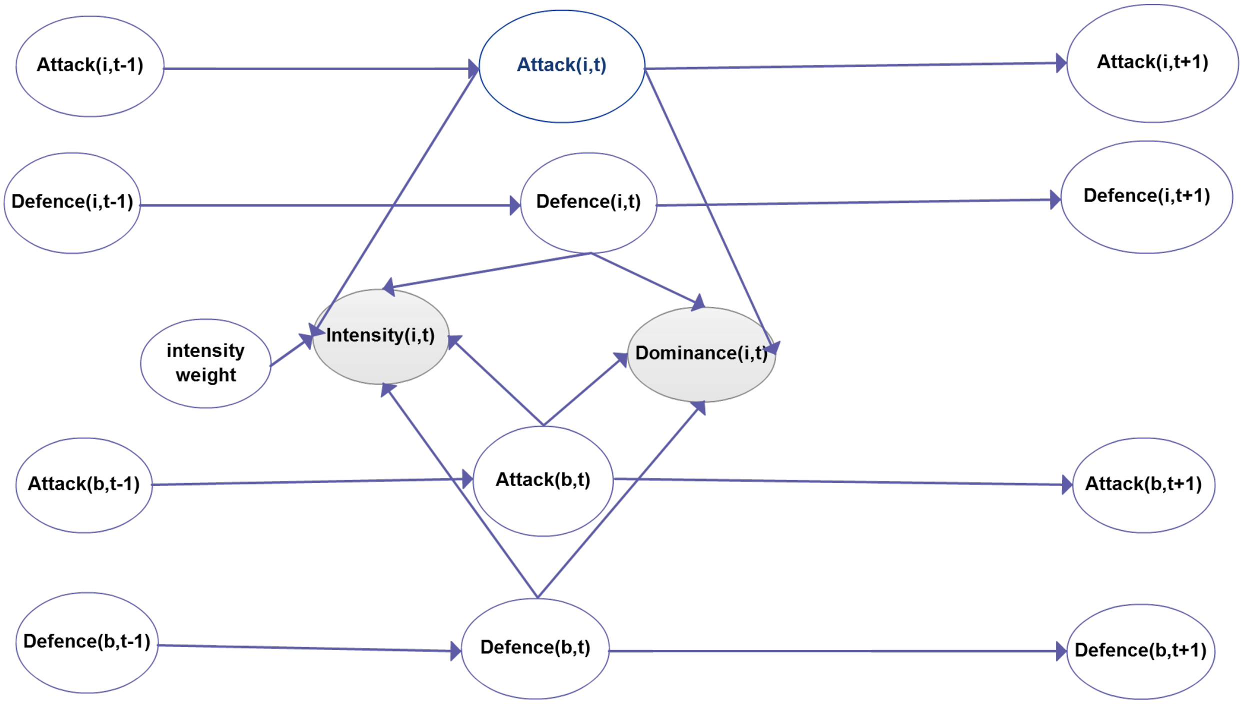


Figure 5.1: Markov chain generated by jags with the extended model

6 Experiments and Results

This chapter starts by giving the experimental plan for how to test the model and compare it with the existing ones of Maher (1982) and Dixon & Coles (1997), stating the experimental setup and presenting the results.

6.1 Experimental Plan

We wish to compare our models ability to predict game results and earn money on betting, compared to those of Maher (1982) and Dixon & Coles (1997). In order to do this, we must use the same data-set on all three models. We will be using data from the English Premier League Season 2012/2013, and we will be starting the betting in round two, and continue until the last round, 38.

We will also be comparing the strengths of the models with different betting strategies; we will again be using the strategy proposed by Rue Salvesen (2000) (the variance-adjusted strategy), explained in the preliminary experiments in section 4.3.2, the fixed bets and -returns methods described in the following subsections

6.1.1 Fixed Bets

A simple strategy where we are placing equally sized bets on all matches, disregarding probabilities and odds (As long as there is a profit to be made, i.e. $probability * odds > 1$).

For our experiment, we will use 2 as the size of each bet.

6.1.2 Fixed Return

Similar to the fixed bets strategy, we will here place bets so that each bet has the opportunity to return a fixed amount. The amount to be placed on a bet is then calculated as :

$$Bet = \frac{Return}{Odds}$$

We will be using a return value of 2 for each bet when using this strategy.

6.1.3 Only Favourites

As Constantinou & Fenton (2013) explain it, there exists a tendency for bettors to place bets on long-odds-outcomes with sub-optimal expected returns, and that bets at short odds tend to generate higher returns.

For example, we may considere a match where a top-team is playing at home against a weak team, and the 'true' probability of a home win is 0.9, and of an away win is 0.05. A bookie may present fair odds for home win, at 1.11 ($1.11 * 0.9 = 1$), while the odds for an away win are at 15 ($15 * 0.05 = 0.75$)[4]. Constantinou & Fenton (2013) state that bettors will yet have a tendency to rather place bets on the risky outcome of an away win, prefering to place small bets with the sight of winning large profits, rather than placing large bets with small profits.

Only favourites is a *true* or *false* parameter which we will use in addition to the other betting strategies. When *true*, Our chosen betting strategy remove two match outcomes, leaving only the short odds outcome remaining. If this outcome yields a positive expected return of investment, the betting strategy will resume as normal, otherwise we will not place bets. When *Only favourites* is used, the betting strategies will therefore always place equal or less amounts on bets than when it is not used.

6.2 Experimental Setup

For each of the models, we use the following constants for the Gibbs sampler:

Thinning: 20. It has already been shown in Section 4.3.3 how a value of 30 is high enough to to a great enough extent reduce the auto-correlation between consecutive samples.

Burn-in: 1000. When first starting, the Gibbs sampler will select random sample-values, and as a consequence of random walks it may take a significant amount of time before it reaches the stationary distribution of the Markov chain. We therefore set the burn-in value to 1000, ignoring the first 1000 samples we obtain for each variable.

Sample-size: We use 10 000 as our sample-size.

Data-set: Match results and statistics from each of the 380 matches played in the 2012-2013 season of the English Premier League.

For the bookmaker odds for each match, we will the odds retrieved from *www.betExplorer.com*, which will be the best available odds for each outcome for each match. The odds are the best odds selected from approximately 40 bookies for each match.

6.3 Experimental Results

Table 6.1 and Figures 6.1 - 6.6 show how the models performed against each other. As we can see, all three models struggled with gaining a profit, and our model never managed to have a net gain during the 38 rounds of the season for either of the six betting strategies used. In fact, only Dixon & Coles (1997) model managed this; a single spike early in the season saw them leap to nearly 40% gain with the variance-adjusted betting strategy, and this same spike is evident in all diagrams presented in Figures 6.1 - 6.6.

However, we may also note that for the final 15 consecutive weeks our model had the highest gain using the variance adjusted betting strategy both when placing bets only on favourites and when not, ending at slightly less than 7% loss. When not using *only favourites*, Dixon & Coles ended up with a 17% loss, while Maher (1982) ended with approximately 10%. Placing bets on only favourites improved Dixon & Coles model by 5%, while reducing the performance of Maher's by 4% over the entire course of the season.

Figures 6.3 and 6.4 shows how the models performed when using the *fixed bet* strategy, with and without the *only favourites* variable. The sum of all bets does not add up to 380∗2 = 760, because there are some matches were neither of the outcomes present favourable odds for our model.

Our model performs slightly worse in both of these scenarios compared to using the variance adjusted strategy, while Maher has a significant drop of 8% when using fixed bets with bets on only favourites. and Dixon & Coles either stay approximately the same as when using variance-adjusted betting without regarding only favourites.

Bet strat.	Maher			Dixon & Coles			Martin			OF
	Bets	Returns	%	Bets	Returns	%	Bets	Returns	%	
Var. adj.	1085.2	967.8	.90	852.3	706.8	.83	909.4	848.5	0.933	No
Var. adj.	1052.6	909.7	.86	646.9	567.8	0.88	860.4	801.5	0.931	Yes
Fix. bet	740.0	669.7	.90	736.0	613.9	.83	732.0	651.4	0.89	No
Fix. bet	402.0	331.2	0.82	318.0	263.7	0.82	438	389.6	0.89	Yes
Fix. ret.	207.5	184.0	0.88	187.0	154.0	0.82	177.0	144.0	0.81	No
Fix. ret.	170.0	144.0	0.84	121.0	104.0	0.86	163.6	150.0	0.916	Yes

Table 6.1: Results of models when using different betting strategies for rounds 2 to 38 accumulated. The last column indicates if the variable *only favourites* was used or not.

Our model performs the worst using the fixed returns betting strategy, with a return of investment of only 81%. However it proves an improvement when placing bets only on favourites, beating both the fixed bet strategies as well, though only very slighty.

From Table 6.2 we see that our model generally performs better than the other two. It is also interesting to note that the roles of the models of Maher and Dixon & Coles have reversed compared to last season; Maher now gives slighty better results.

Model	Average Accuracy
Maher	.866
Dixon & Coles	0.84
Martin	0.893

Table 6.2: The average return for each model over the six betting strategies used.

Considering the figures 6.1 - 6.6, it is a recurring feature that our model performs worst of the three in the opening 10-15 rounds, and from then on noticeably outperforms both Maher and Dixon & Coles.

Team	FLP (points)	Predicted FLP(points)	Diff.
Manchester United	1 (89)	1(75.51)	0
Manchester City	2 (78)	2(66.37)	0
Chelsea	3 (75)	3(66.16)	0
Arsenal	4 (73)	4(65.58)	0
Tottenham	5 (72)	5 (63.37)	0
Everton	6 (63)	6 (60.24)	0
Liverpool	7 (61)	10 (52)	3
West Brom	8 (49)	7 (58.79)	1
Swansea	9 (46)	9(53)	0
West Ham	10(46)	13(49.15)	3
Norwich	11(44)	11(51.81)	0
Fulham	12(43)	16 (43.52)	4
Stoke	13(42)	8 (57.17)	5
Southampton	14(41)	15(44.82)	1
Aston Villa	15(41)	18(41.02)	3
Newcastle	16(41)	14(47.32)	2
Sunderland	17(39)	12(50.73)	5
Wigan	18(36)	17(42.26)	1
Reading	19(28)	19 (34.52)	0
QPR	20(25)	20 (33.46)	0

Table 6.3: The final league table as predicted by our model, after half the matches have been played. Average error in position is 1.4

Table 6.3 Shows the actual final league positions (FLP) of each team, and then the predicted league table generated by our model. Though the standard deviation of the point distribution is not as high as in the actual table (the distance in points between the title winner, Manchester United, and QPR, who finished last, is only approximately 42 points in our model's prediction, whereas in reality it is 64), the model manages to predict correctly

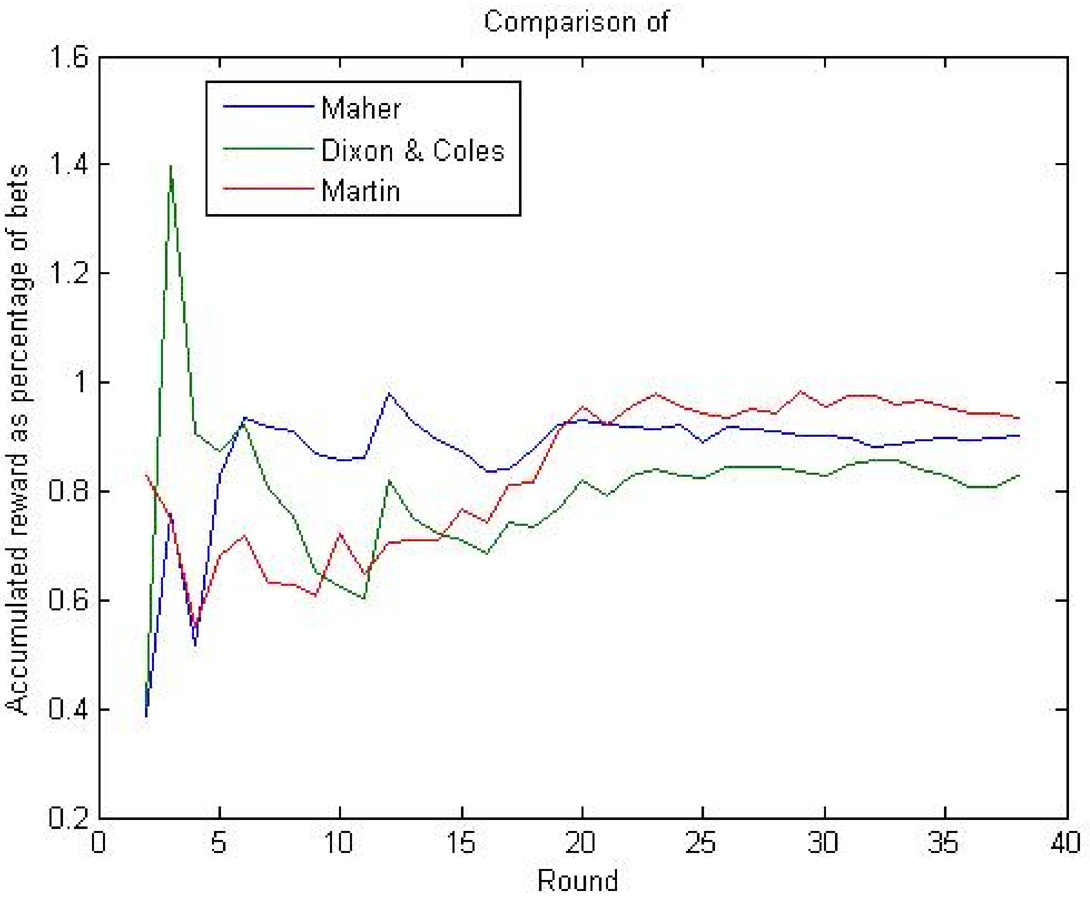


Figure 6.1: Accumulated returns of the three models between rounds 2 and 38 when using variance-adjusted betting strategy.

the positions of 10 of the 20 teams in the league, with an average error of 1.4 in positioning.

We may also note that though Wigan seemed to be performing well with regards to their *intensity*, *dominance* and *shots on target* values presented in tables 5.6, 5.7 and 5.8 (being placed 6th, 9th and 9th, respectively in these regards), our model still managed to place the team within a single position from their final standing.

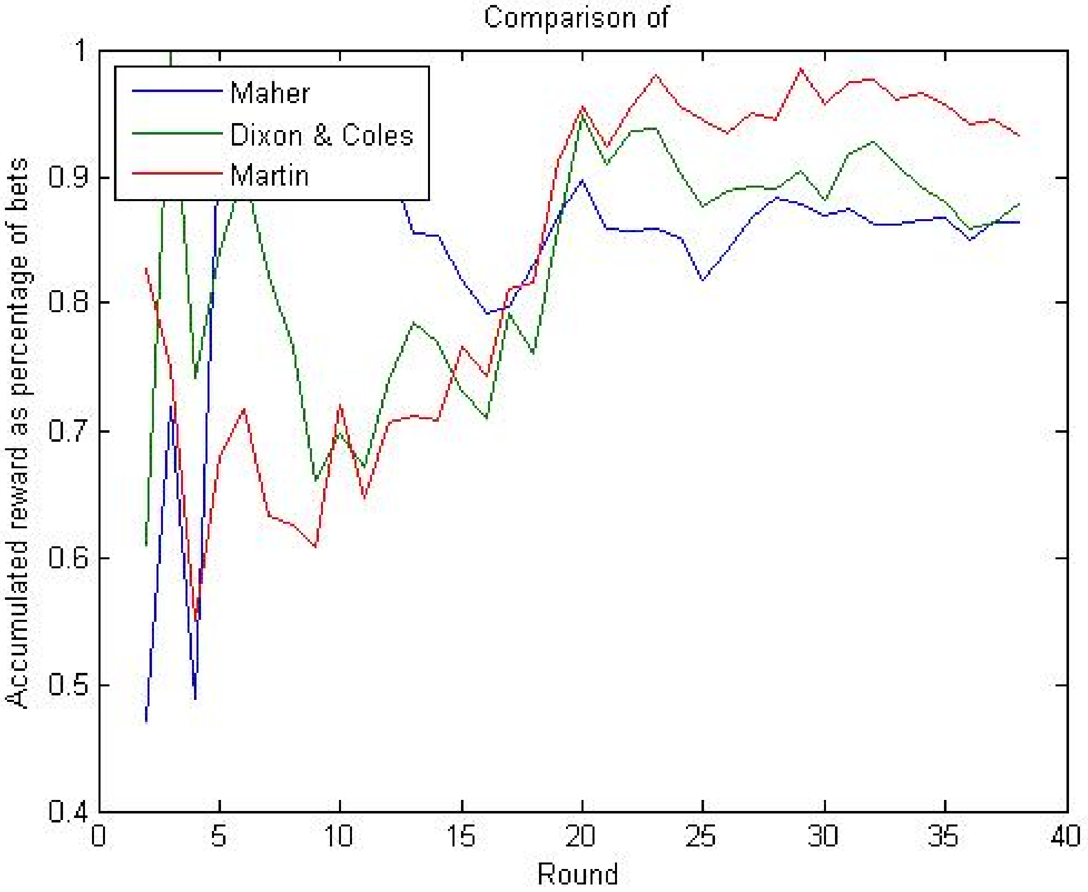


Figure 6.2: Accumulated returns of the three models between rounds 2 and 38 when using the variance-adjusted betting strategy and betting only on favourites.

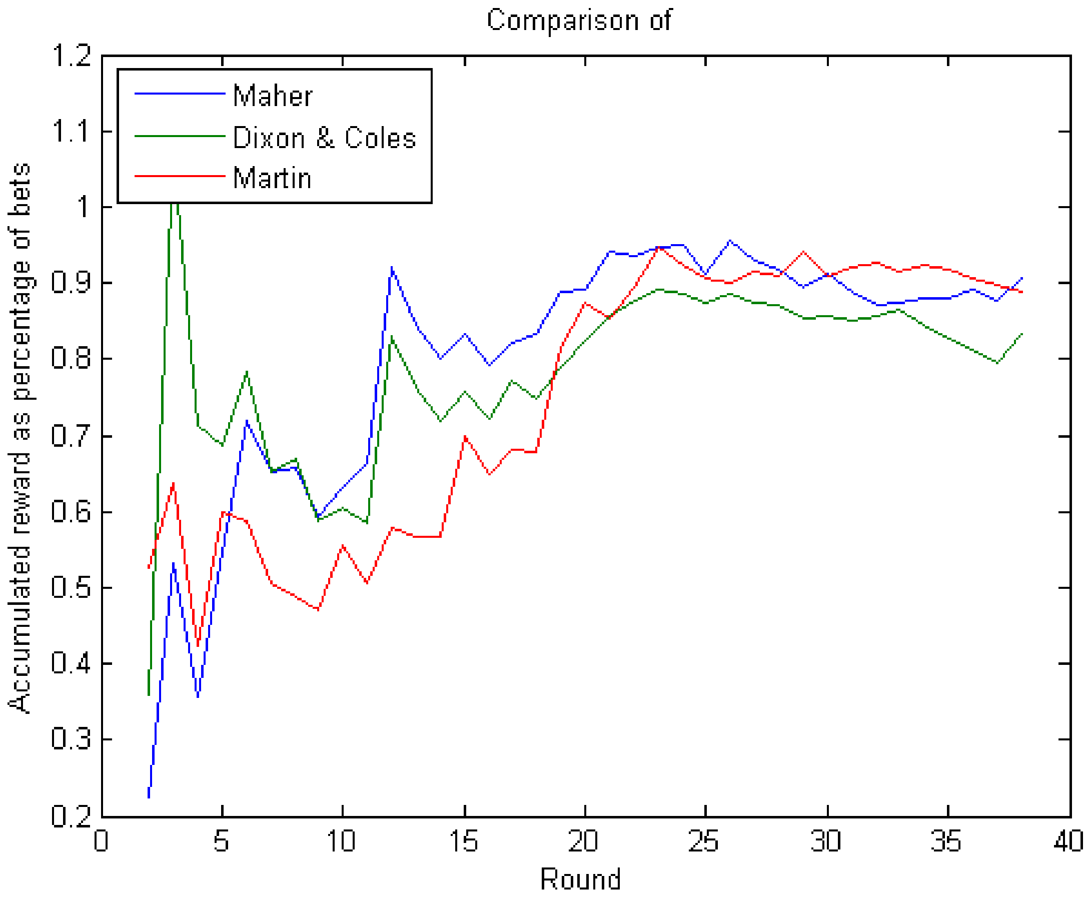


Figure 6.3: Accumulated returns of the three models between rounds 2 and 38 when using fixed bet betting strategy.

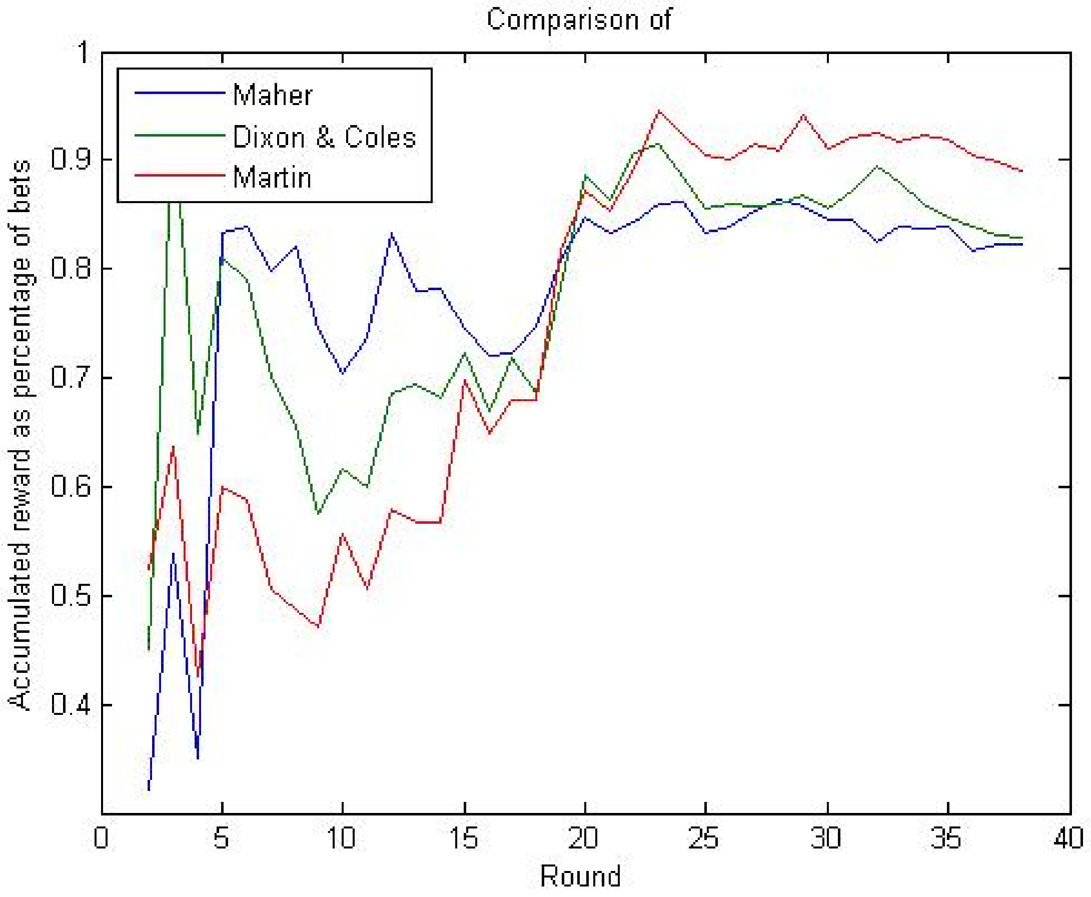


Figure 6.4: Accumulated returns of the three models between rounds 2 and 38 when using the fixed bet betting strategy and betting only on favourites.

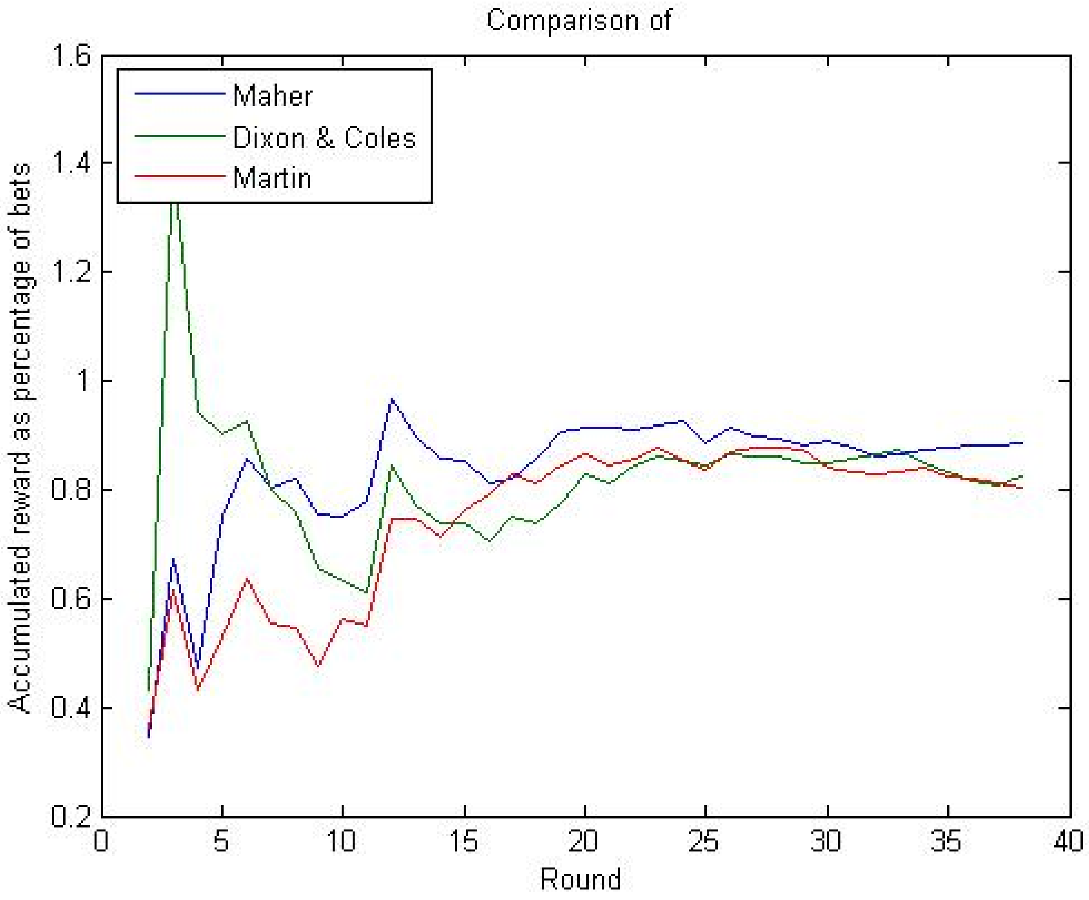


Figure 6.5: Accumulated returns of the three models between rounds 2 and 38 when using the fixed return betting strategy.

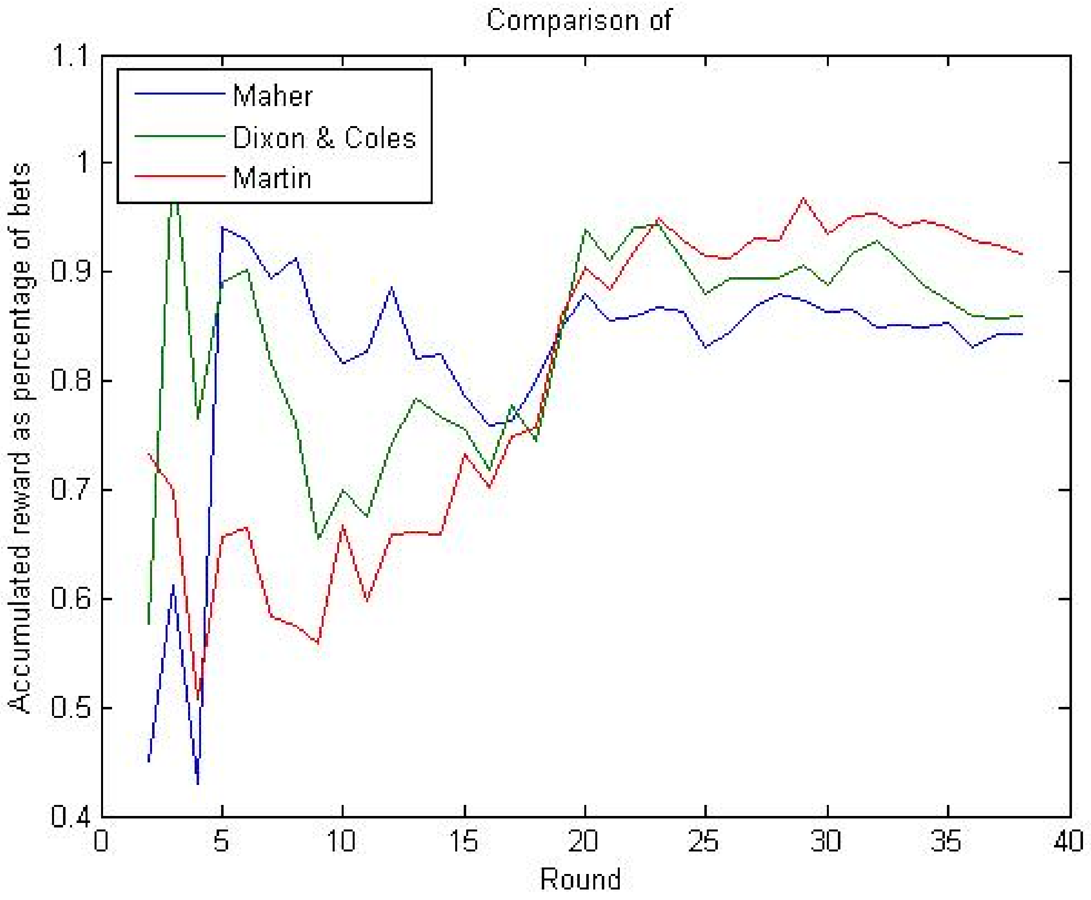


Figure 6.6: Accumulated returns of the three models between rounds 2 and 38 when using the fixed return betting strategy and betting only on favourites.

7 Evaluation and Conclusion

In this chapter we evaluate the results from the research and experimentation, followed up by some details on what to continue to work on with regards to this project. The continuation of work is a synthesis of work mentioned in the preliminary study of this project as well as newly obtained findings.

7.1 Evaluation

From the results of the previous chapter, it is not so easy to determined that we have successfully managed to produce a model which more accurately assesses the probability of outcomes in upcoming matches. Though it does not have as high a return of investments as neither Maher (1982) nor Dixon & Coles (1997) had during the English Premier League season 2011/2012, it outperforms both on a regular basis when using the newest league data available. However, the fact that it still does not manage to produce a profit yields opportunity for further research on the subject. Also when regarding how Maher (1982) worked better than Dixon & Coles (1997) for this season, while not for the previous one, we may question our results. The performances are so close that any differences may be negligable, as last season the difference between the models of Maher and Dixon & Coles was 10% in returns on invenstment, and we did not manage to achieve differences of that magnitude in our experimental results.

W may take note that our model seems to especially perform better when all three models focus only on favourites in each match. This may indicate that our model is slightly better at assessing the probability of the favourite winning.

As we also saw in the experimental results, our model performs well when predicting the final league table and positions of individual teams.

7.2 Discussion

At the beginning of this thesis we described two research areas we wished to investigate, given in section 1.2. in this section we discuss how well our research questions have been answered.

7.2.1 Useful in-depth data

One of the questions we wished to answer, were to assess which data proved the most useful when attempting to predict future results. Our findings in chapters 5 and 6 indicate that the amount of shots a team managed to get on target give the best indication of how the result of a match ended. When attempting to predict goals a team will score in the following match, the intensity, a composite variable consisting of possession-,tackling- and heading-statistics provide the best results.

7.2.2 Improvement of model

We wanted to assess whether adding more data would improve our model, receiving a higher return of investments when betting on matches. The results from chapter 6 show that our model seems to perform slightly better than those it is compared against, but is unable to perform better than the bookmakers. It seems that the model is still suffering from too general match data. If we could, for instance, assess how many of each shot on target were products of 'clear-cut' chances, and how many were shots from distance or poor angles, we may be able to explain some of the under and over-achievers. TOn the other hand, placing shots in such categories may be deemed subjective matter, as 'clear-cut' chances are interpreted differently by different people, and how far away must a shot be to be considered a 'long-shot'? What if the goal keeper is off his line?

7.3 Contributions

By continuing the development of the simulator designed by our supervisor, Helge Langseth, we have contributed in the area of applying more in-depth statistics from matches in order to predict outcomes of football matches. Previous work has focused on using final scores of matches, whereas we have applied data which give a deeper understanding of how previous matches have unfolded, and how well teams have actually performed.

We have also developed an internet crawler that can retrieve this match-data from any league, as long as the data exists for that league, as soon as matches have been played.

7.4 Continuation of Work

We may take note of how our model seems to perform considerably better during the second half of the seaso, as evident of figures 6.1 - 6.6. It would be interesting to investigate how our model would perform for next season if we were to include the data we already have obtained from this season:

Using data from previous years: Consider trying to estimate team strengths at the time of the first round of a season for a given league. We have no data to use for this estimation, and our model will therefore be inaccurate at the start of the season, increasing in accuracy as the season progresses. However, including data from previous seasons is not as straight forward as may first be assumed. Each season, three teams will have been relegated, and three teams promoted from a division (this is not alwas the case, but is for the English Premier League, and the exact number of teams is beside the point). For any year, some teams were thus not in that league the previous year, and so we cannot use a complete model of the season preceeding the one we are scrutinizing, as these teams do not appear in that data. It is not quite as simple as to just add the data for the three new teams, as this will have been for either a division higher or lower than the one we are interested in, and so these estimates would be misguiding. A team relegated from a higher division will assumably have poor estimates of either attacking or defensive skills, presumably both, but it may nevertheless be one of the strongest teams in the division below.

Player transfers: Similar to the previous point, simply carrying a team's strengths over from one season to the next, will not capture the importance of key transfers made between clubs in the off-season in between. For example, many rate Luis Suarez and Gareth Bale to have been vital parts of their teams Liverpool and Tottenham, respectively, this season. In fact both were nominated for the PFA Player of the Year award this season [1],having both scored more than 20 goals this season. Both are now at the start of the transfer window rumoured to be leaving their current clubs. The impact this would have on the attacking power of Tottenham and Liverpool is of course difficult to quantify, but some impact it will most probably have. It could deem profitable to have a model which takes note of such transfers.

Identification and availability of important players: If we were able to obtain data on starting line-ups of which players are playing for each team, together with data on who scores each goal in every match, we could build an algorithm which reduces the probability of a team winning based on the fact that "important" players are missing. A player which scores often, or a large portion of the team's goals, can be considered important. Playmakers may also be categorized this way if we can obtain information on which players are second-to-last on the ball when the team scores. Defenders may be given a defensive importance meter by recognizing that when certain players play, the team generally concedes fewer goals than when others are playing.

Importance of avoiding relegation and winning the league: Goddard & Asimakopoulos (2004) calculate whether a team has a chance of being relegated or promoted from a league, and adjust their strengths accordingly. From their results it seemed that their models improved nearing the end of the seasons, which may indicate that this is a variable which has a positive effect on the income generated by the model, as relegation and promotion opportunities become more evident towards the end of the season.

Days since previous match: Counting the days between matches for a team may yield important information about the qualities of the team. Too few days since the last match may indicate that the team must rotate their starting line-up, giving the opposition a slight edge into the game. This would coincide with the *availability of important players* variable, but could also give a pointer to the general fatigue of the team.

Season betting We see in table 6.3 that our model is accomplished at predicting final league positions of teams, correctly identifying the first 6 positions. Obtaining odds for season betting, where we bet on final league positions of teams may be an interesting direction to take the model, as stronger teams have higher *intensity* and *dominance* values on average over the season.

Some of these variables are possible to obtain, being presented in the same source code as where we find our current match statistics. This includes players used for each match, and who scored. However using them could require a restructuring of our current setup for data-files.

Bibliography

[1] Bloom, B. (2013), ”Gareth Bale, Luis Suarez and Michael Carrick included in PFA Player of the Year nominations”, The Telegraph 19.04.2013, accessed 03.06.2013 at `http://www.telegraph.co.uk/sport/football/competitions/premier-league/10005694/Gareth-Bale-Luis-Suarez-and-Michael-Carrick-included-in-PFA-Player-of\-the-Year-nominations.html`

[2] Cattelan, M., Varin, C. and Firth, D. (2012), Dynamic Bradley–Terry modelling of sports tournaments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, not yet published.

[3] Constantinou, A.C. and Fenton, E. (2012), Solving the Problem of Inadequate Scoring Rules for Assessing Probabilistic Football Forecast Models. *Journal of Quantitative Analysis in Sports*, 8,1-12,

[4] Constantinou, A.C. and Fenton, E. (2013), Profiting from arbitrage and odds biases of the European football gambling market. *Submitted for publication.* Accessed 06.06.2013 at `http://constantinou.info/downloads/papers/evidenceofinefficiency.pdf`

[5] Crowder, M., Dixon, M., Ledford, A. and Robinson, M. (2002), Dynamic modelling and prediction of English Football League matches for betting. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 51: 157–168.

[6] Dixon, M.J. and Coles, S.G. (1997), Modelling Association Football Scores and Inefficiencies in the Football Betting Market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46, 265–280.

[7] Dixon, M. and Robinson, M. (1998), A birth process model for association football matches. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47, 523–538.

[8] Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (1995), *Markov Chain Monte Carlo in Practice*, London: Chapman & Hall/CRC Interdisciplinary Statistics.

[9] Goddard, J. and Asimakopoulos, I. (2004), Forecasting football results and the efficiency of fixed-odds betting. *J. Forecast.*, 23, 51–66.

[10] Goddard, J. (2006), Who wins the football?. *Significance*, 3, 16–19.

[11] Grayson, J. (2012), "Another post about TSR" accessed at 04.06.2013 at http://jameswgrayson.wordpress.com/2012/07/15/another-post-about-tsr/

[12] Hirotsu, N. and Wright, M. (2003), An evaluation of characteristics of teams in association football by using a Markov process model. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52, 591–602.

[13] Karlis, D. and Ntzoufras, I. (2003), Analysis of sports data by using bivariate Poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52, 381–393.

[14] Knorr-Held, L. (2000), Dynamic Rating of Sports Teams. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49, 261–276.

[15] Maher, M.J. (1982), Modelling association football scores. *Statistica Neerlandica*, 36, 109–118.

[16] Matlab interface (2013), "MATJAGS 1.3 A Matlab interface for JAGS", accessed 04.06.2013 at http://psiexp.ss.uci.edu/research/programs_data/jags/.

[17] McHale, I. and Scarf, P. (2007), Modelling soccer matches using bivariate discrete distributions with general dependence structure. *Statistica Neerlandica*, 61, 432–445.

[18] Parsing script (2008), "PHP Simple HTML DOM Parser", accessed 04.06.2013 at simplehtmldom.sourceforge.net

[19] PHP Introduction (2013), "What is PHP?", accessed at 07.06.2013 at http://www.php.net/manual/en/intro-whatis.php

[20] Read CSV-type spreadsheats that have values that are not all numeric (2011), "read_mixed_csv.m" accessed 06.06.2013 at http://stackoverflow.com/questions/4747834/matlab-import-csv-file-with-mixed-data-types

[21] Rue, H. and Salvesen, O. (2000), Prediction and Retrospective Analysis of Soccer Matches in a League. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49, 399–418.

[22] Russel, S. and Norvig, P. (2010), *Artificial Intelligence: A Modern Approach*, 3rd edition, New Jersey: Pearson Education, Incorporated.

[23] WampServer (2013), "Wampserver: a Windiws web development environment", accessed 03.06.2013 at http://www.wampserver.com/en/

Appendix A

A.1 League Positions 2011/2012

The two following figures show how the league positions of each team in the English Premier League changed over the course of the season 2011/2012. Wigan Athletic is outlined in red.

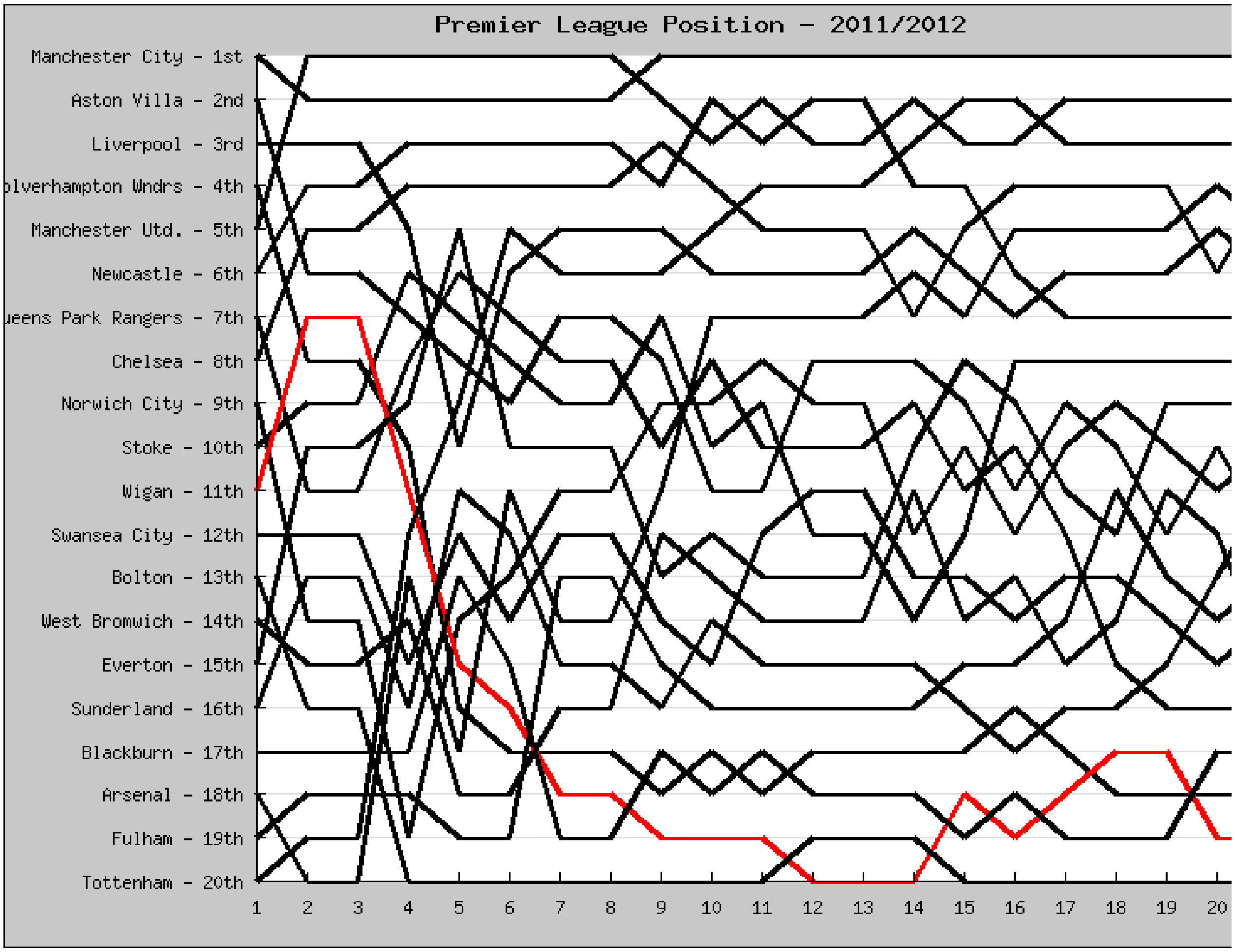


Figure A.1: League positions of each team in the Premier League, season 2011/2012, part 1 (Wigan outlined in red)

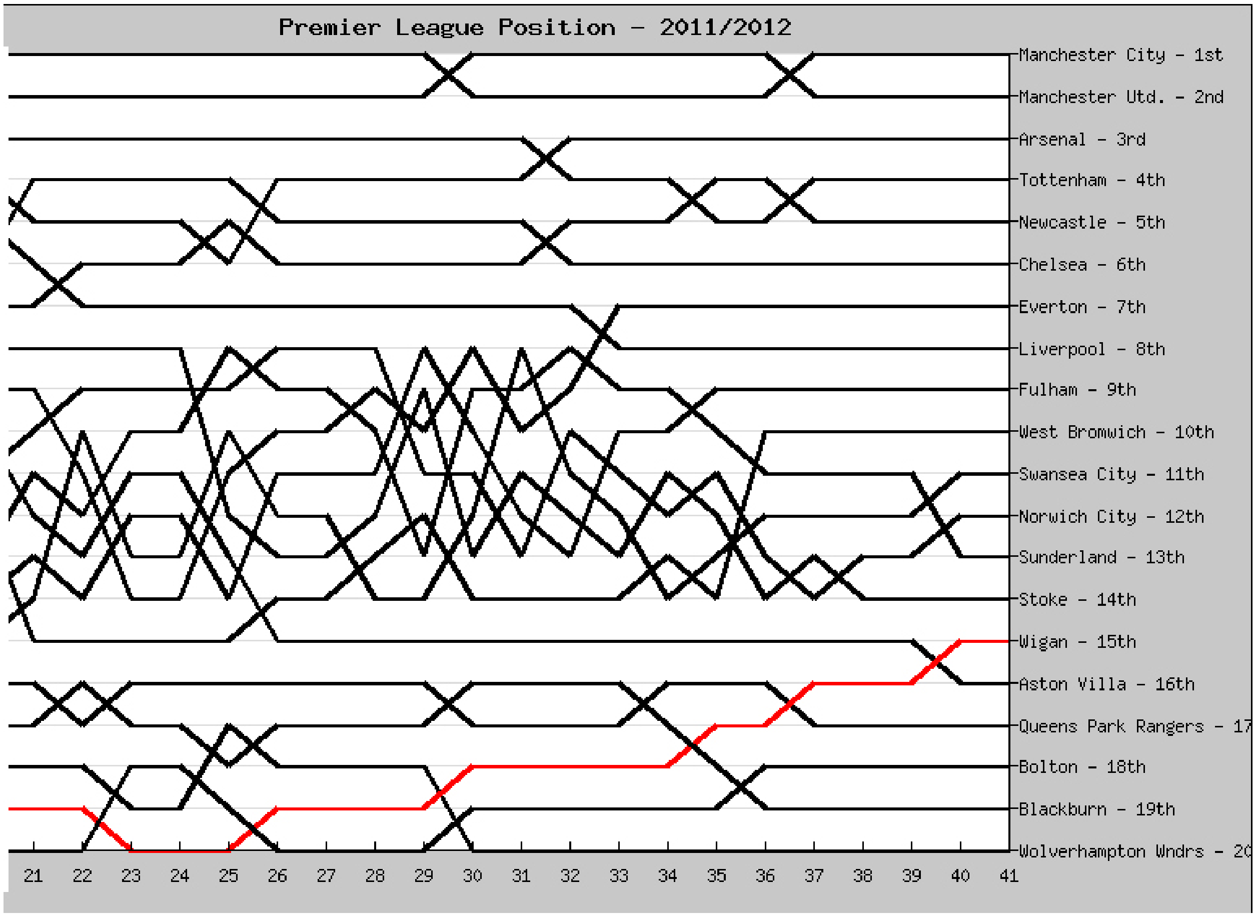


Figure A.2: League positions of each team in the Premier League, season 2011/2012, part 2 (Wigan outlined in red)

A.2 Maher model implementation

In this section we give you the full implementation of the Maher (1982) model for the JAGS program, written by Helge Langseth.

```
1.  model {
2.    goalLambda ~ dgamma(1, 1)
3.    awayDisadvantage ~ dunif(0, 1)
```

Lines 2-3 assign the two laten variables *goalLambda* and *awayDisadvantage* their prior distributions. *awayDisadvantage* is a multiplication constant used for scaling number of goals scored by a team when playing away-games. We scale the attacking and defending abilities so that the sum of all attacks are equal to the sum of all defences. This is done by forcing the each vector to sum to *noTeams*, which is 20. *GoalLambda* takes care of the offset.

We proceed with setting prior distributions for the unscaled attack and defence values for each team.

```
4.    # UNSCALED abilities
5.    for (i in 1:noTeams){
6.      # unscaled abilities
7.      unscaledDefence[i] ~ dunif(0,1)
8.      unscaledAttack[i] ~ dunif(0,1)
9.    }
```

The next section deals with setting the values for the scaled versions of defence attack. The average value for both defence and attack will be 1.

```
10.    # SCALED abilities
11.    for (i in 1:noTeams){
12.       defence[i] <- noTeams * unscaledDefence[i] / sum(unscaledDefence)
13.       attack[i] <- noTeams * unscaledAttack[i] / sum(unscaledAttack)
14.    }
```

Lines 15-29 sets the goals scored for home and away sides in each of the 380 matches, using the Poisson distribution with the mean for each distribution as described in section 4.1.

```
15.    # And then, the goal-model
16.    # Starting to loop over all games
17.    for (game in 1:noGames){
18.       # Home team
19.       goalsScored[game, 1] ~ dpois(
20.             goalLambda * attack[schedule[game, 1]]
21.             * defence[ schedule[game, 2] ]
22.            )

23.      # Away team
24.      goalsScored[game, 2] ~ dpois(
25.            awayDisadvantage * goalLambda *
26.            attack[schedule[game, 2]] * defence[ schedule[game, 1]]
27.              )
28.    }
29.  }
```

A.3 Dixon and Coles model implementation

In this section we give you the full implementation of the Dixon & Coles (1997) model for the JAGS program.

```
1.  data {
2.    for (game in 1:noGames) {
3.     ones[game] <- 1
4.    }
5.    C <- 100
6.  }
```

Lines 1-6 are part of the "ones-trick". JAGS can only handle a set of standard distributions, such as the Poisson or the Bernoulli distributions. We are interested in tweaking the Poisson distribution the way Dixon & Coles (1997) did, and therefore use the ones-trick. This is done by tricking the Gibbs sampler into thinking that it is drawing randomly, while we in fact make sure that it samples a 1 each time. Using the property of the Bernoulli distribution with mean x, where the likelihood of observing a 1 is the mean , x. Thus, we can obtain our desired probability distribution using it as x in the Bernoulli distribution. Lines 69-74 complete the ones-trick.

```
8.  model {
9.     goalLambda ~ dgamma(1, 1)
10.    awayDisadvantage ~ dunif(0, 1)

# Define temporal model
#########################
15.    for (i in 1:noTeams){
16.    # First timestep
17.      unscaledDefence[i, sortedGame[i,1]] ~ dunif(0, 1)
18.      unscaledAttack[i, sortedGame[i,1]] ~ dunif(0, 1)

20.      # ...  and the looping
21.     for (t in 2:noRounds) {
22.      unscaledDef[i,sortedGame[i,t]] ~ dnorm(unscaledDef[i,sortedGame[i,t-1]],1)
23.      unscaledAtt[i,sortedGame[i,t]] ~ dnorm(unscaledAtt[i,sortedGame[i,t-1]],1)
26.     }
25.   }
```

In lines 8-10 we start set up some initial values for two of the latent variables. Lines 15-18 we give initial values to the unscaled attack and defence parameters for each team, in each round. For the first round, they are picked from the uniform distribution. For the rest of the rounds, each team's unscaled attack and unscaled defence are determined by sampling from the normal distribution, where the mean is the unscaled attack and defence for the given team from the previous round. The *sortedGame[i, t]* array ensures that we pick the correct round in case some matches have been moved due to some unforseen event (snowstorm or other weather conditions, etc.).

```
# we make a temp variable which at each point is the sum of all teams' unscaled
# abilities up to that point

28.   for(t in 1:noRounds){
29.    tempDef[1, t] <- unscaledDef[1, sortedGame[1, t]]
30.    tempAtt[1, t] <- unscaledAtt[1, sortedGame[1, t]]
31.    for(i in 2:noTeams){
32.     tempDef[i, t] <- unscaledDef[i, sortedGame[i, t]] + tempDef[i-1, t]
33.     tempAtt[i, t] <- unscaledAtt[i, sortedGame[i, t]] + tempAtt[i-1, t]
34.    }
35.   }
```

For getting the scaled defences and attacks, we need to divide the unscaled attack or defence of each team on the sum of all the unscaled attacks or defences, based on the round we are currently looking at. the language used in JAGS interprets variables as constants, so that we are not allowed to overwrite a variable such as tempDef inside a for-loop. Instead we must create an array which holds all the temporary accumulated unscaled defence and attack values (tempDef and tempAtt). When the inner for-loop on line 31 is done executing, tempDef[number of teams, round] will hold the accumulated value of all the defence values for the given round.It is then a simple matter to calculate the scaled attack and defence values (*attack* and *defence*) for each team, making it easier to compare strengths of each team.

```
37.   for ( i in 1:noTeams){
38.    for(t in 1:noRounds){
39.     defence[i, sortedGame[i,t]]
40.      <- noTeams * unscaledDef[i, sortedGame[i, t]] / tempDef[noTeams, t]
41.     attack[i, sortedGame[i,t]]
42.      <- noTeams * unscaledAtt[i, sortedGame[i,t]] / tempAtt[noTeams, t]
43.    }
44.   }
```

```
# And then, the goal-model
# Starting to loop over all games
47.  for (game in 1:noGames){

# Home team
49.   lambda[game] <- goalLambda * attack[schedule[game, 2], schedule[game, 1]]
50.     * defence[ schedule[game, 3], schedule[game, 1]]
51.   goalsScored[game, 1] ~ dpois( lambda[game] )

# Away team
53.   mu[game] <- awayDisadvantage * goalLambda
54.     *attack[schedule[game, 3], schedule[game, 1]]
55.     * defence[ schedule[game, 2], schedule[game, 1]]
56.   goalsScored[game, 2] ~ dpois( mu[game] )
```

In lines 47- 56 we set up the λ and μ values for each team, for each round. the number of goals scored for the hometeam (goalsScored[*game*, 1]) and awayteam (goalsScored[*game*, 2]) in the match with ID *game*, is sampled using the Poisson distribution with their respective means λ and μ.

```
# For defining the rho
58.   lowerBound[game] <- max( -1 / lambda[game], -1 / mu[game] )
59.   upperBound[game] <- min( 1, 1 / (lambda[game] * mu[game]) )

# Declare some variables to find the tau-scaler
61.   is0X[game] <- ifelse(goalsScored[game, 1]==0,1,0)
62.   isX0[game] <- ifelse(goalsScored[game, 2]==0,1,0)
63.   is1X[game] <- ifelse(goalsScored[game, 1]==1,1,0)
64.   isX1[game] <- ifelse(goalsScored[game, 2]==1,1,0)
65.   is00[game] <- is0X[game] * isX0[game]
66.   is01[game] <- is0X[game] * isX1[game]
67.   is10[game] <- is1X[game] * isX0[game]
68.   is11[game] <- is1X[game] * isX1[game]

69.   ones[game] ~ dbern( (is00[game] * ( 1 - lambda[game]*mu[game]*rho ) +
70.              is01[game] * ( 1 + lambda[game]*rho ) +
71.              is10[game] * ( 1 + mu[game]*rho ) +
72.              is11[game] * ( 1 - rho ) +
73.              1-(is00[game]+is01[game]+is10[game]+is11[game]))/C )
74.   } # closing for-loop starting at l. 47

76.   rho ~ dunif( max(lowerBound[]), min(upperBound[]) )
77.  } # closing model starting at l. 8
```

Lines 58-59 Set up the boundaries for ρ as described in section 3.2. Lines 61-68 are used for making the ones-trick more readable. Lines 69-74 utilizes the ones-trick to tweak the Poisson distribution by τ, here given as the mean for the bernoulli distribution. For each value of *game*, only one of the values *is00[], is10[], is01[]* and *is11[]*can be 1, as they are mutually exclusive. Hence multiplication in the lines 69-73 ensures that only one of the parts will be applied as the mean of the bernoulli distribution.

The program will execute this model once for every sample we want. The amount of samples is specified in the MATLAB framework and passed to the JAGS program through the matjags() method.

A.4 Our model implementation

In this section we give you the full implementation of the model we have designed for the JAGS program.

```
1.  data {
2.    for (game in 1:noGames) {
3.     ones[game] <- 1
4.    }
5.    C <- 100
6.  }

7.  model {
8.    goalLambda ~ dgamma(1, 1)
9.    awayDisadvantage ~ dunif(0, 1)
10.   tauObserve ~ dgamma(1,1)
11.   tauDynamic ~ dgamma(1,1)
12.   intensityWeight ~ dgamma(1,1)
```

Lines 7-12 we start up some intial values for five latent variables. *TauObserve* and *tauDynamic* are used as precision in the distributions of *attack*, *defence*, *intensity* and *dominance.* This way we do not set a predifined precision, but let JAGS approximate the value.

```
13.   for (i in 1:noTeams){
14.    dominance[i, sortedGame[i,1]] ~ dnorm(attack[i,sortedGame[i,1]]
15.      -defence[i,sortedGame[i,1]]
16.      -(attack[opposition[i,sortedGame[i,1]],sortedGame[i,1]]
17.      -defence[opposition[i,sortedGame[i,1]],sortedGame[i,1]]),tauObserve)
18.    intensity[i, sortedGame[i,1]] ~ dnorm(intensityWeight
19.      *(attack[i,sortedGame[i,1]]
20.      -defence[i,sortedGame[i,1]]
21.      -(attack[opposition[i,sortedGame[i,1]],sortedGame[i,1]]
22.      -defence[opposition[i,sortedGame[i,1]],sortedGame[i,1]])),tauObserve)
23.     attack[i,sortedGame[i,1]] ~ dnorm( 0, tauDynamic )T(0, )
24.     defence[i, sortedGame[i,1]] ~ dnorm( 0, tauDynamic )T(0, )

25.     for (t in 2:noRounds){
26.       dominance[i, sortedGame[i,t]] ~ dnorm(attack[i,sortedGame[i,1]]
27.         -defence[i,sortedGame[i,1]]
28.         -(attack[opposition[i,sortedGame[i,1]],sortedGame[i,1]]
29.         -defence[opposition[i,sortedGame[i,1]],sortedGame[i,1]]),tauObserve)
30.       intensity[i, sortedGame[i,t]] ~ dnorm(intensityWeight
31.         *(attack[i,sortedGame[i,1]]-defence[i,sortedGame[i,1]]
32.         -(attack[opposition[i,sortedGame[i,1]],sortedGame[i,1]]
33.         -defence[opposition[i,sortedGame[i,1]],sortedGame[i,1]])),tauObserve)
34.       attack[i, sortedGame[i,t]] ~
35.         dnorm(attack[i, sortedGame[i,t-1]],tauDynamic)T(0, )
36.       defence[i, sortedGame[i,t]] ~
37.         dnorm(defence[i, sortedGame[i,t-1]],tauDynamic)T(0, )
38.       }   #Closing for-loop starting at line 25.
39.     }     #Closing for-loop starting at line 13.
```

Lines 13-39 initiate the missing data variables of *intensity* and *dominance*, and also the latent variables *attack* and *defence*. The T(0,) at the end of the declarations of *attack* and *defence* are for truncating the normal distributions so that all values below 0 is set to zero. This is so that the poisson distributions used in lines 46 and 51 will work.

```
40.   for (game in 1:noGames){
41.
42.       # Home team
43.       lambda[game] <- goalLambda
44.         *attack[schedule[game, 2], schedule[game, 1]]
45.         * defence[ schedule[game, 3], schedule[game, 1]]
46.       goalsScored[game, 1] ~ dpois( lambda[game] )

47.       # Away team
48.       mu[game] <- awayDisadvantage * goalLambda
49.         * attack[schedule[game, 3], schedule[game, 1]]
50.         * defence[ schedule[game, 2], schedule[game, 1]]
51.       goalsScored[game, 2]   dpois( mu[game] )

53.       lowerBound[game] <- max( -1 / lambda[game],
54.         -1 / mu[game] )
55.       upperBound[game] <- min( 1, 1 / (lambda[game]
56.         *mu[game]) )

57.       # Declare variables to find the tau-scaler
58.       is0X[game] <- ifelse(goalsScored[game, 1]==0,1,0)
59.      isX0[game] <- ifelse(goalsScored[game, 2]==0,1,0)
60.      is1X[game] <- ifelse(goalsScored[game, 1]==1,1,0)
61.      isX1[game] <- ifelse(goalsScored[game, 2]==1,1,0)
62.      is00[game] <- is0X[game] * isX0[game]
63.      is01[game] <- is0X[game] * isX1[game]
64.      is10[game] <- is1X[game] * isX0[game]
65.      is11[game] <- is1X[game] * isX1[game]

66.      ones[game] ~ dbern( (is00[game]
67.         * ( 1 - lambda[game]*mu[game]*rho ) +
68.         is01[game] * ( 1 + lambda[game]*rho ) +
69.         is10[game] * ( 1 + mu[game]*rho ) +
70.         is11[game] * ( 1 - rho ) +
71.         1-(is00[game]+is01[game]+is10[game]+is11[game]))/C )

72.   }
73.   rho ~ dunif( max(lowerBound[]), min(upperBound[]) )

74.  }
```

Lines 40-51 are identical to lines 47-56 of the Dixon & Coles (1997) model, which is described in the previous subsection. Lines 1-6 and 53-74 performs the ones-trick already described in the previous section dealing with the Dixon & Coles (1997) model.

Appendix B

B.1 Structured Literature Review

This section aims to give a structured literature review. It will assess the need for the review before proceeding with specifying research questions which will later try to be answered. To filter out literature which is not relevant, there will be stated a review protocol, which defines the rules used to determine what literature will be used in the research, and what will not. Finally there will be given a set of terms used for searching, and a list of sources where the terms will be used.

B.1.1 Rationale

The objective of this literary review is to examine which solutions currently exist in the field of building betting machines that predict the outcome of football matches, and examine how well these currently perform. It will focus on the differences and similiarities in models used by the different predictors, as well as the assumptions made in order to simplify methods or make the methods work, and the feasability of these assumptions.
There are many aspects that make the need for a literary review evident. A structured literary review helps place this work in a more historical perspective, and presents the current state of the art of the domain in which the thesis lies. It also helps prevent the possibility of redoing the work and findings of others, while promoting further research on others' results.

B.1.2 Research Questions

In order to begin research, it is important to have set some goal for what the research should try to answer, as well as some research questions for focusing on certain aspects of existing solutions to the problem. These research questions will later also be used to assess which works should be excluded or included in the structured literature review.

Research Question 1: What existing solutions are there for predicting football match outcomes?

It is important to find several alternatives solutions to the same problem. This will make it easier to find strengths and weaknesses in the solutions provided, and from there build our own solution, which uses the most successful aspects of earlier research done. Finding existing solutions also prevents us from having to build a whole solution ourselves, but rather use information gathered by others before us.

Research Question 2: How do the existing solutions aquired by question 1 compare to each other with regards to models or data used?

There may be very different ways in which team strengths are presented. Whether they are vectors, integers or double values, it is important to find out whether these representations matter, either in regards of computational time required, or simply for making models more comprehendable.

The existing solutions will probably use different data sets for experimentation, whether it is from the year 1990 or 2007, or the English Premier League or Norwegian Tippeligaen, all depending on where the authors may be from and when the paper was written. Have the models been tested on several different datasets? A modified Poisson distribution of goals scored per match may be very suitable for a given league in a certain year, but how well does it fit for some other data?

Research Question 3: What empirical evidence is there to suggest that some of the models found by Question 2 produce better betting predictions than others ?

We assume that for the existing solutions to the problem, there will have been done tests which present the predictive ability of the given solution, and whether or not it provides a positive income when used together with a betting strategy. There are, however, many betting strategies one may follow when testing a model, and also many ways to assess the accuracy of predictions (such as Geometric mean, Information Loss and Maximum Log-likelihood). This question is concerned with what conclusions we may make regarding the empirical evidence of the different models. How can we be certain that the differences between the rewards earned by two opposing models is due to the strengths of the models themselves, and not the different betting strategies used?

B.1.3 Review Protocol

In this section, we describe the review protocol used to assert whether a discovered literary object is deemed relevant.

The studies should match the inclusion criterias, as well as the quality criterias. The studies should not match the exclusion criteria.

B.1.4 Key Terms and Strings

Provided below is the strategy for how sources will be searched through. The table below shows the terms used, which together form different search strings. Each term is divided into several words which are synonyms of each other, or provide related semantic meaning. Each key term is in its own group.

ID	Criteria Description
Inclusion 1	The study's main concern is match prediction or determining football team ratings
Inclusion 2	The study presents empirical results in the fields of prediction or team ratings
Exclusion 1	The study is mainly concerned with American Football
Exclusion 2	The study's main concern is with live-betting on matches
Exclusion 3	The study is mainly concerned with knock-out tournaments
Quality 1	The study's aim is clearly stated

Table B.1: List of inclusion criteria

When searching a source, every permutation of a group will be tried out with every permutation of every other group, resulting in a list of literature which contains one synonym from each group (or best match).

	Group 1	Group 2	Group 3
Term 1	predict	football	match
Term 2	model	soccer	scoring
Term 3	analysis		result
Term 4	determine		league
Term 5	assess		rating

Table B.2: list term groups used for searching sources

B.1.5 Sources

Below are the sources which will be used for finding state-of-the-art literature in the field of football prediction. The strings built by using the key terms in section 2.2.4 will be used for searching these databases. In total these terms build 50 unique search strings.

The Wiley Online Library publishes several journals known to produce relevant articles in the field of football match prediction and betting, such as the Journal of Royal Statistical Society: Series C and D, Statistica Neerlandica and the Journal of Forecasting. Thus, searching through the Wiley Library of journals means simultaneously searching through the mentioned journals of interest, as well as unknown journals that might have relevant papers on the subject.

There are also some interresting journals that the Wiley Library does not publish, and these will be listed below and searched through as well.

- Journal of Quantitative Analysis in Sports
- Journal of Statistical Computation and Simulation
- Wiley Online Library

B.1.6 Selected Primary Studies

Method

By using the Wiley Online Library "advanced search" method, sets of 10 strings were used simultaneously to search through the database. These sets would be of the form "predict football match" OR "predict football result" OR... etc. This procedure was repeated 5 times in order to use all possible strings from section 5. Each string was used to search the *abstracts* of the articles only, not the entire articles.

Using 10 strings at a time, there were five sets of 30 + 90 + 110 + 70 + 54 = 354 results, though there was a high percentage of overlap. For the vast majority of the articles found, reading the abstract was enough to concude that neither Inclusion Criteria 1 nor 2 had been met, and the articles were therefore discarded.

The Taylor and Francis Online Library was used to search through the Journal of Statistical Computation and Simulations. However, there was no way here of quickly searching through the journal as done with Wiley. Attempting to search for the single term "football" and subsequently "soccer" in the articles of this journal produced three and seven results, respectively. None of these met Inclusion Criterias 1 or 2. As all strings to be used would return subsets of these two results, the journal was discarded as a source for relevant papers.

After a set of papers was obtained through these methods, each article's text went through a screening process where the whole text was analyzed to verify whether it met the criterias stated in section 2.2.3.

Results

This section first provides the set of articles that passed through the abstract-level screening, and then gives a explanation of why some of the articles were removed from the set in the full-text screening.

Abstract screening

Using the sources provided, the search terms stated and filtering on the review protocol given, the set of studies which passed the abstract-level screening is the collection of the following articles:

- Cattelan, M., Varin, C. and Firth, D. (2012), Dynamic Bradley–Terry modelling of sports tournaments. *Journal of the Royal Statistical Society: Series C (Applied Statistics).*
- Constantinou, A.C. and Fenton, E. (2012), Solving the Problem of Inadequate Scoring Rules for Assessing Probabilistic Football Forecast Models. *Journal of Quantitative Analysis in Sports.* Volume 8, Issue 1.

- Crowder, M., Dixon, M., Ledford, A. and Robinson, M. (2002), Dynamic modelling and prediction of English Football League matches for betting. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 51: 157–168.

- Dixon, M. J. and Coles, S. G. (1997), Modelling Association Football Scores and Inefficiencies in the Football Betting Market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46: 265–280.

- Dixon, M. and Robinson, M. (1998), A birth process model for association football matches. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47: 523–538.

- Goddard, J. and Asimakopoulos, I. (2004), Forecasting football results and the efficiency of fixed-odds betting. *J. Forecast.*, 23: 51–66.

- Goddard, J. (2006), Who wins the football?. *Significance*, 3: 16–19.

- Hirotsu, N. and Wright, M. (2003), An evaluation of characteristics of teams in association football by using a Markov process model. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52: 591–602.

- Karlis, D. and Ntzoufras, I. (2003), Analysis of sports data by using bivariate Poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52: 381–393.

- Knorr-Held, L. (2000), Dynamic Rating of Sports Teams. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49: 261–276.

- Maher, M. J. (1982), Modelling association football scores. *Statistica Neerlandica*, 36: 109–118.

- McHale, I. and Scarf, P. (2007), Modelling soccer matches using bivariate discrete distributions with general dependence structure. *Statistica Neerlandica*, 61: 432–445.

- Rue, H. and Salvesen, O. (2000), Prediction and Retrospective Analysis of Soccer Matches in a League. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49: 399–418.

Full-text screening

By reading the articles in full-text, some of the articles were found to be lacking in relevance, as they had a too much different focus compared to our project. The articles removed from the study, are:

- **Goddard, J. (2006), Who wins the football?**
  This article is not concerned with prediction of football matches, but analyzes results from English football from the past 30 years, to try to put objective facts and statistics

to back up or debunk subjective opinions. It does not provide a method for modelling team ratings. It presents some empirical evidence in areas such as the the influence of recent European cup matches (Champions League or Europa League) on domestic performances, evidence of the home advantage the in the last 30 years, and the role of geographical distance when playing away games. However, we will not be using any European match history in our modelling, and at a closer look on the study done by Goddard (2006), some of his conclusions do not have any basis in the tables he provides.

Goddard (2006) states that the greater the distance the away team has to travel, the poorer is the away team's average performance. However, the table clearly shows a steadily increasing percentage of away-wins the further the team must travel. Also, the average amount of goals scored increases with the distance traveled for the away team, and decreases for the home team.

- **Dixon, M. and Robinson, M. (1998), A birth process model for association football matches.**
  This article is concerned with the prediction of football matches, and provides a model to assess the probabilities of the possible outcomes, but does so with respect to live-betting.

  The model proposed uses the general spread of goal-times over the approximately 90 minutes a match lasts, as well as the empirical data found that more goals are scored later in the game, and that the amount of goals scored up to a given point in a game influences the probability of more goals, to determine the outcome of a live match. As we are to predict matches before they start, this has no value for us. Removing the live-betting aspect by using the model for the time t=0 at the start of the match, removes the need for these variables, but also makes the model more general, which does not add much to other existing models.

B.1.7 Quality Assessment

For each of the remaining papers, an evaluation of the quality must take place. This will be done by giving points to each paper according to how well they fulfill the statements in B.3 below. If the paper fulfills a statement, it will receive 1 point, if it only partially fulfills it, a half point is given. Finally, no points are received if a paper does not fulfill a given statement.

Articles with higher score than 5 will be kept, while those with a score below 5 will be discarded.

Nr	Description
1	The study is put into context of other studies and research
2	The model used is reproducable
3	The test data set reproducable
4	The algorithms used are explained in detail and thoroughly elaborated
5	The results are thoroughly analyzed
6	The conclusion is consistent with the results presented

Table B.3: Quality assessment statements

Article scores

This section gives the individual scores of each paper, in the form:

Article author (Year) = **Total points** *(Points from statement 1 + points from statement 2 + ... + Points from statement 6).*

Any paper with a score below 5 of 6 points will not be deemed a suitable study, and subsequently removed from the set.

Cattelan, M., Varin, C. and Firth, D. (2012) = **6**(1+1+1+1+1+1)

Constantinou, A. C. and Fenton, E. (2012) = **6**(1+1+1+1+1+1)

Crowder, M., Dixon, M., Ledford, A. and Robinson, M. (2002) = **5.5**(1+1+1+1+0.5+1)

Dixon, M. J. and Coles, S. G. (1997) = **6**(1+1+1+1+1+1)

Goddard, J., and Asimakopoulos, I. (2004) = **6**(1+1+1+1+1+1)

Hirotsu, N. and Wright, M. (2003)= **6**(1+1+1+1+1+1)

Karlis, D. and Ntzoufras, I. (2003) = **5.5**(1+1+1+1+0.5+1)

Knorr-Held, L. (2000) = **5**(0.5+1+1+1+0.5+1)

Maher, M. J. (1982) = **6** (1+1+1+1+1+1)

McHale, I. and Scarf, P. (2007)= **5.5**(1+1+0.5+1+1+1)

Rue, H. and Salvesen, O. (2000) = **6**(1+1+1+1+1+1)

B.2 State of the Art Assessment

This section presents the contributions others have made into the field of football prediction and team strength assessments. Here we will systematically go through the different main approaches provided in other research.

B.2.1 Using the Poisson Distribution

Maher (1982) describes how early work in the field of modelling team strengths and predicting football match outcomes used the negative binomial distribution to model the amount of goals a team would score during a given match. Maher (1982) states that this assumption erroneously implies that all teams have equal strengths, contradicted by works that show how football league final standings may be quite correctly predicted by experts. This seems a strong indication that Maher (1982) may be right in his assumption that chance may play a considerable role in a single match, but over several matches it dissipates, overshadowed by the differences in team abilities.

Maher (1982) provides an interpretation of a football match which gives rise to a binomial probability of number of goals, approximated by the Poisson distribution. Maher (1982) assumes that each time a team has possession of the ball, it has the opportunity to attack, which may subsequently result in a goal. With n attacks during a game, with the probability p of an attack resulting in a goal, the number of goals can be approximated by the Poisson distribution. This requires the assumption that the probabiltiy p for a goal is constant and each attack independent. Though using the Poisson distribution has provided many good results, it may be questioned whether these are reasonable assumptions. One may argue that an attack starting with winning possession of the ball from the opposition goalkeeper yields a higher p value than an attack starting with tackling the opposition striker in your own penalty-area. An attack leading to one team going into a 3-0 lead may have negative effects on both teams' p value, or even n, as securing the win by keeping possession of the ball becomes a higher priority, leading to fewer attacking opportunities for both teams.

B.2.2 Introducing the bivariate Poisson model

Utilizing the Poisson distribution, Maher (1982) uses the product of the home-team's attack ability and away-team's defending ability as the mean amount of goals scored by the home-team, and vice versa for the away'team. Maher (1982) first assumes the goals scored by each team are independent of each other, before improving his model by introducing a bivariate Poisson model. When using the independent models for home- and away-goals, Maher (1982) states that they can be interpreted as two separate games at each end of the pitch, which may be a over-simplification of the game. This is also demonstrated by the bivariate Poisson version of the model, which improves the results considerably. The correlation between goal scoring at the two ends of the pitch has been debated in several papers. Karlis & Ntzoufras (2003) argue that the bivariate Poisson model should be used, as it is reasonable to assume that since the two teams interact with one another, the two outcome variables of scores are correlated. Continuing, one team increasing their game speed in search for a goal will lead to more goal-scoring opportunities for both teams. Using data from the Champions League from season 2000-2001, Karlis & Ntzoufras (2003) show that a hypothesis of no correlation between goals scored was rejected. However, McHale & Scarf (2007) provide statistics from the English Premier League during the period August 2003 to March 2006, and according to their findings, there seems to be little evidence of any correlation between goals scored by the opposing teams. This makes it difficult to pinpoint why exactly the model provided by Maher (1982) improved when adding dependencies between the teams' scoring, and whether correlation is too dependent on the sample set used. We may take note that the Champions League of one season may be a small set, certainly much smaller than that of three full Premier League seasons.

Karlis & Ntzoufras (2003) also considered, as Maher, using the bivariate Poisson distribution to model team capabilities, and shows how increasing the correlation between goals scored, has a positive effect on the prediction of amount of games drawn. Using a correlation factor of 0.2, as Maher (1982) used, gives a 14% increase in expected number of draws. Karlis & Ntzoufras (2003) further improve their model by using similar inflating methods as Rue & Salvesen (2000) and Dixon & Coles (1997), but rather than inflating the results 0-0 and 1-1, Karlis & Ntzoufras (2003) inflate the probabilities of draws in general. Karlis & Ntzoufras (2003) states that inflating the probabilities of draws turns the marginal distributions into mixtures of distributions with a single Poisson component, thus maybe correcting the overdispersion of results. It may be noted that the sample set used in this paper is taken from the Italian Serie A season 1991-1992, a time where victories were rewarded with 2 points, not 3, as in more recent times. This reduced reward for winning games would quite understandably decrease a teams will to push on for the win. Consequently, samples from this period would most likely contain generally more drawn results than those from a newer time. This may lead to the model produced by Karlis & Ntzoufras (2003) being less applicable for predicting todays matches.

B.2.3 The Home Ground Advantage and Varying Team Form

By testing increases in Maximum Log Likelihoods when adding parameters to his model, Maher (1982) found that introducing individual attacking and defending abilities increased the accuracy of his model. Meanwhile, adding parameters to describe a team's strengths when playing away games did not increase the likelihood of the model at the 1% level, and Maher (1982) concluded that it is enough to add a constant factor for all teams to provide for the advantage that comes with playing at home. Maher (1982) does not provide any details on the origins of the home-advantage effect and whether or not it is a fair assumption, but Dixon & Coles (1997) show that for the period 1993-1995 and over 6000 matches in English football, the ratio of outcomes are 46% home wins, 27% draws and 27% away wins, which provide enough evidence for this to be a valid assumption to make. Cattelan et al. (2012) also showed that for the 2008-2009 season in the Italian Serie A, an average of 65% of points each team accumulated over the season, was obtained in home-games. Knorr-Held (2000) provides data from the 1996-1997 season of the German Bundesliga, showing that of all the games played, 51% ended in home wins, and only 26% resulted in away wins. Put in context to each other, these findings strongly indicate that whichever footballing league one uses data from, and whenever these data are from, there seems to be an inherent advantage to the team playing at home, for whichever reason.

Where Maher (1982) it seems assumed that each team's strength was constant over the period of a season, several others have later attempted to use dynamic attacking and defending abilities of teams to capture the variable performances a given team may have over the season. Though this may seem a subjective opinion, and that variations in a team's (superficial) performance may be caused by chance, there are several reasons why chance may not have the only say in this: Dixon & Coles (1997) and Knorr-Held (2000) state how performance in a particular game may be influenced by the ability of newly arrived players, changing of the coach, unavailability of injured players or sacking of a manager. This seems reasonable, as removing or including an essential part of a team may easily alter the overall strength of that team. One must however also consider that adding the possibility of variable team strengths may lead to an overfitting of the model, where a few wins followed by a couple of losses leads to the team being interpreted as first one of the best teams in it's league, then quickly reduced to one of the worst. Rue & Salvesen (2000) use a parameter t to indicate how far back in time we will look to find match results used to estimate the team's current strength. Each match then has a decreasing influence on the team's current ability, as we move further away from it in time.

Crowder et al. (2002) also refine the model introduced by Dixon & Coles (1997), proposing a more elaborate scheme for evolving team strengths and weaknesses over time. They also avoid using the MCMC approach as it is deemed more computationally demanding, providing instead an approximating method which is concluded to be competitive with the original Dixon & Coles (1997) method, however slightly poorer at predicting matches. As computing capacity has been improving since the dawn of computers, certainly since the publishing of this paper, the methods considered less tractable by Crowder et al. (2002) seem far more feasible now, making this approximation approach slighty redundant.

B.2.4 Altering the Poisson Distribution by Inflation

Dixon & Coles (1997) build upon the approach taken by Maher (1982), introducing a simple approach to such a fluctuation in team capabilities. They too stick to using each team's history of match scores alone, in order to estimate strengths. Rather than building on the conclusion of Maher(1982) that the bivariate Poisson model provided better results, Dixon & Coles (1997) use the initial independent assumption of goals by the two opposing teams. They find that the independent model is particularly bad at predicting the scorelines 0-0, 1-0, 0-1 and 1-1, and that the bivariate Poisson distribution does not sufficiently improve these results. Hence the model is modified to improve the expected amount of the mentioned four outcomes, while keeping the marginal distributions of goals scored by teams X and Y Poisson. There are however more score-lines which the Poisson distribution either over- or under-estimates; 4-3, 3-4, 3-3, and 6-1 are all results which suggest the independence between scores is unreasonable, but the modified Dixon & Coles (1997) model does not take these into account. One may argue that these results are relatively rare compared to those that the authors adjust for, and because of this, it takes only a few occations too many or few in the sample set to make the model seem unreasonable.

Rue & Salvesen (2000) use Bayesian methods to update time-dependent estimates of team strengths each time a new match has been played, and the Markov chain Monte Carlo (MCMC) techniques are iteratively used for inference of simultaneous, dependent abilities of all teams in a league. Rue & Salvesen (2000) use Dixon & Coles (1997)' modified independent Poisson model, further altering it in two significant ways. They observe that as the scores become high, the probabilities divert from the Poisson distribution. Rue & Salvesen (2000) propose to adjust the model to this observation by truncating the the results, interpreting all scores above 5 as 5. They state that any scores beyond this amount does not give any added information on the teams' abilities. This implies not only reading the score line 6-3 as 5-3, but also 7-5 as 5-5. Though such results may not occur very often, one may question what impact reading a win as a draw may have on each of the teams' abilities, and whether that impact is a correct assumption. It might be better to keep the goal difference, and reduce both teams' scoring rate until it reaches an acceptable level, for instance reducing the result 7-5 to 5-3. This way we remove the excess goals that add little information, while leaving the result unaltered.

Rue & Salvesen (2000) also assume that each match is less informative for team abilities than the modified Poisson model might suggest. They alter the model in order to imply that at least a part of each match result can be forecasted by looking at the average amount of goals scored for the league. This seems appropriate, as for instance in a league where the average team scores 1 goal each game, scoring 1 goal in a game may reduce the effect this has on altering the current ability of a team. Their findings show that assuming approximately 80% of a match result yield information, provides the best predictive results.

B.2.5 Being Superior

As the only authors, Rue & Salvesen (2000) include the psychological effect of a match between two teams that differ largely in quality. Rue & Salvesen (2000) assume that within the same league, a strong team will tend to underestimate the weaker team, leading to a closer match-up then initially predicted. The parameter is assumed a small constant, equal for all teams in the league. The authors do not seem to have any legitimate basis for this parameter to exist, but empirical evidence suggests that a value of 0.1 may give the best results. These results may have been further improved by including the home-advantage factor, which Rue & Salvesen (2000) do not use. This seems a slight oversight, as the home-advantage may be a more important factor than the proposed psychological effect, and also a more recurring one.

B.2.6 Further Research on Team Characteristics

Hirotsu & Wright (2003) are more concerned with understanding the characteristics of football teams, rather than using previous results to establish the strengths of each team. In contrast to many others (Maher (1982), Karlis & Ntzoufras (2003) and Rue & Salvesen (2000)) who use mainly scoring distributions for evaluating teams, Hirotsu & Wright (2003) focus on more detailed information from games,such as the amount of passes made and overall possession over the course of a game. Hirotsu & Wright (2003) propose a Markov process model for the transitions between having and giving away possession of the ball, and between having possession of the ball and scoring a goal. The authors also test the goodness of fit of the Poisson distribution for the expected frequency of both number of goals and also number of gains in possession, and could not find any significance at the 5% level, thus not rejecting the Poisson assumption. This is constistent with earlier works, which also assume a Poisson distribution. It should be mentioned that Hirotsu & Wright (2003) do not attempt to build a model dependent on both a team's ability to score goals and retain possession of the ball, but keep the two interpretations of team skill separated. It would be interesting to see if there were any correlation between a team's ability to keep possession of the ball, and amount of goals scored. No empirical results are reported of the performance of the model, though, overall, for both scoring, conceding, gaining and losing possession, the higher ranked teams receive better marks than those farther down the league table.

McHale & Scarf (2007) apply bivariate Poisson-related distributions to the amount of shots each team make during a game, and show how they are negatively dependent. As the authors try to seek out a bivariate discrete distribution, they consider the copula functions which generate such distributions. Using Frank's copula, the authors find a distribution which models the observed shots taken by home and away teams well. McHale & Scarf (2007) find that there is many factors that seem to have an effect on how many shots a team manages to get during a match. Key findings according to the authors, are that away teams in general are more efficient in producing goal scoring opportunities, as home teams need more passes and crosses in order to produce shots. Commiting fouls also have a larger negative impact on chances created for the home team than the away side.

B.2.7 Other Models

Knorr-Held (2000) employed a cumulative link model assuming random-walk priors for abilities of teams. The variance of the random walk is estimated through four different predicitive criteria. The abilities of individual teams are estimated by means of an extended Kalman filter and smoother algorithm. The author uses a single rather than two values to describe the strength of a team; Where most others previously determine attacking and defending strengths of a team, Knorr-Held (2000) simply has a parameter a to determine it's ability. This author also, similar to Maher(1982) amongst others, assumes additional threshold parameters, such as the home advantage, are time constant and team independent. A problem with the model that Knorr-Held (2000) proposes, is that there may not exist any maximum likelihood estimates for team abilities if a team has only experienced losses or wins. This, as Knorr-Held (2000) explains, would lead to either positive or negative estimated ability. This would prove problematic if attempting to assess abilities of teams at an early stage of a season. In many leagues, it is not unlikely that a few teams may during the first five or so rounds experience only wins or losses, and under such circumstances, the Knorr-Held (2000) model would not be applicable.

Goddard & Asimakopoulos (2004) use an ordered probit model to determine which covariates affect outcomes of matches. Such covariates may be yellow or red cards dealt, number of fouls, geographical distance between teams, importance of a match with regards to final league-table standings, specifically matches which may have a direct impact on which teams are relegated or promoted or win the championship. The latter seems an especially important factor, and one that other papers have failed to assess: The difference between two teams may be less evident when the least skilled team has a larger ulterior motive to win the game, and the better team does not. Similarly, the probability of a better team winning against a mid-table team may be further increased if it is known that the better team has a chance at the Championship, while the mediocre team no longer has a chance of neither being champions nor being relegated to a lower division, and predictive models should try and capture this notion.

The results of the betting done by Goddard & Asimakopoulos (2004) shows that their model only in some areas produce positive gross returns, but on a whole, is not able to outperform the bookmakers. However, for both the 1999 and 2000 season of English football leagues, there is positive generated return of 8% each year for the April-May period, signifying the end-of-season games. We can assume that this is when the Significance-parameter kicks in for each team, and may be an important factor in this profit, though Goddard & Asimakopoulos (2004) do not mention this. It does seem that there is an exploitable inefficiency in the betting market during this period of the season, as the authors state.

Cattelan et al. (2012) use the Terry-Bradley model for assessing team strengths, further modifying it by constructing a dynamic version which takes into account the changing strength of team's over time. When evolving the ability of a team over time, Cattelan et al. (2012) state that for calculating home-game abilities of a team, they only consider matches the team has previously played at home ground, not taking into account perhaps more recent matches if they were away games. The same applies for calculating away game strengths. This seems a slight over-simplification, as it may be reasonable to assume that recent games have an impact on the next game, wherever they play. Two straight away wins may for instance have a great psychological effect on a team going into a match on their home grounds. The authors also take into account all previous home-games found in the sample-set when calculating the dynamic home-game strength of a team, whereas Rue & Salvesen (2000) for example have found that only using matches up to 100 days old yield the most promising results with regards to prediction. One may question how much significance may be placed on matches that happened far back in time, but as the weighting is of an exponential nature, matches far back are given very little weighting.

B.2.8 Using Inadequate Scoring Rules

Constantinou & Fenton (2012) do not provide a model for forecasting match results or estimating team strengths, but shed light on some of the inadequacies of most scoring rules often used when validating or comparing model forecasting accuracies. The authors explain how football outcomes are ordinal, and that forecasting models may be easily validated using the Rank Probability Score. Their conclusion raises the concern that many of the findings of previous researchers may be invalid, as they have used scoring rules that do not under all circumstances identify the most accurate model as thus.

Constantinou & Fenton (2012) give five example matches with accordingly two different prediction sets for each match, A and B and show how the most used scoring rules rank the two given forecasters. The most controversial of these matches is the last one, where all of the scoring rules Geometric mean, Information Loss and Maximum Log-Likelihood- at least one of which are used by both Dixon & Coles (1997), Rue & Salvesen (2000), Hirotsu & Wright (2003) and Karlis & Ntzoufras (2003)- wrongly conclude that the second of the two forecasters, B, is the best. However, looking more closely at match 5, which resulted in a home win, it can be seen that the second model actually has i higher probability of a a home win (B sets the probability of a home win at .60, compared to .57 of A). Constantinou & Fenton (2012) argue that A is the better one, for betting systems where one is certain that the home team will *not lose*, as A has a higher combined probability of home win *or* draw. The mentioned researchers that use the scoring rules that classify the second model as the best, does not use this form for betting systems, and so this fifth match becomes redundant, as neither models of Dixon & Coles (1997), Rue & Salvesen (2000), Hirotsu & Wright (2003) or Karlis & Ntzoufras (2003) would place a bet on the away team not winning, i.e. the result [H *or* D]. This removes some of the concern regarding previous research, but it may still be important when considering various constants that earlier researchers have found give the best results, for instance the suggestions of Rue & Salvesen (2000) for the parameters indicating the importance of the psychological effect of being superior, or how long ago a match can be played before it has no significance for the next match to be played.

B.2.9 Comparision of Models

The different models proposed are difficult to compare as there is a vast sample space of data which a researcher may use for building their model, and it only continues to grow as time passes and more football matches are played. Data used varies between the years 1970 to 2007, and different leagues have also been used, such as English, German and Italian. It is then problematic to assess whether one model has classified the strength of Bayern Munchen in the German Bundesliga of the 1996-1997 season in a better manner than someone else classified Manchester United in the English Premiership in 2006-2007. This may be easierly done when the researchers proceed to use these models in a betting environment to predict future matches.

Not all of the research done provide empirical tests of models, where the model has been used to predict matches, and been applied to betting strategies. Rue & Salvesen (2000) provide a betting strategy of betting on outcomes with positive expected profit, while at the same time keep the variance in profit low. They also attempt combination betting, where they tried predicting three matches at a time. This proved less successfull, but they managed to get a profit when placing single bets, though the lower bound of the variance in the results indicated there was still some risk in losing money. Dixon & Coles also attempted a betting strategy with their model, and provide results which are borderline significantly larger than the return expected with random betting coupled with the standard bookmaker's take. The variance is, however, as in the case with Rue & Salvesen (2000), very large, and a definite conclusion is difficult to make. Goddard & Asimakopoulos (2004) also attempt a betting strategy, and overall end up with a small profit, most noticably performing well during the start and end of both the 1999 and 2000 season.

Cattelan et al. (2012) use the Ranked Probability Score (RPS) to validate the accuracy of their model, which overall scored a mean over all the matches predicted of 0.451. As Constantinou & Fenton (2012) explain it, lower scores of RPS mean a better ability to predict outcomes. As the mean score for Cattelan et al. (2012) is below 0.5, we may assume it is a decent model. Maher (1982) is more interested in examining how well his model predicts the number of goals in matches, which it does quite well, than predicting actual outcomes in matches. For instance, he examines the count of expected number of matches in which team A scores 1 or 2 goals, or team B scores 1 or two goals, or the difference in goals is -1, 0 or +1, and compares these to the observed counts of such events. He does not, however, attempt to examine if there is an overlap in observed and expected events, for instance, did a game that ended 2-2 also be predicted as 2-2, or a draw? Similarly, Karlis & Ntzoufras (2003) also compare many different models and their ability to estimate drawn matches, but only in a very general manner. The authors find that a bivariate Poisson model which has been modified by inflating probabilities for all draw outcomes, most correctly mirrors the distribution of the sample set they use, but it is unknown how well this model actually predicts outcomes of individual matches. Crowder et al. (2002) test their model by assessing the distribution of their predictions, for instance, For all games observed as home wins, the model predicted 48% of these correctly as home wins, and incorrectly predicts 28% as draws and 24% as away wins. The result is worse for observed away wins, correctly predicting 39%

as away wins.

Neither McHale & Scarf (2007) nor Hirotsu & Wright (2003) attempt to use their models for forecasting, but use them only to examine and explain certain characteristics of teams in the data sets they use.

Both Cattelan et al. (2012) and Rue & Salvesen (2000) seem to agree that it would be reasonable to assume that including data beyond simple match results (scores) to their models would result in more accurate data fitting and improved forecasts.

Appendix C Code Documentation

This appendix describes the most important functions and classes in both the internet crawler written in PHP used for information gathering as well as the MATLAB-simulator. It will also present examples of what is stored in each of the datafiles used.

C.1 Data files

This section provides examples of how the *Comma-Separated-Values* (.csv) files should look like in order for our system to work. The section goes through the files chronologically, i.e. in the order they should be updated when using the internet crawler.

Lines in several of the data-files are too 'wide' to be presented in it's actual form in this document. Line numbers will be used to indicate that text is on the same line.

C.1.1 rawTable.txt

```
1.  DataStore.prime('standings', { stageId:  6531 },
1.  [[6531,32,'Manchester United',1,38,28,5,5,86,43,43,89,1,19,16,0,3,
1.  45,19,26,48,1,19,12,5,2,41,24,17,41,
1.  '<a class="d a" id="615224" title="West Ham 2-2 Manchester United"/>
1.  <a class="w h" id="615270" title="Manchester United 3-0 Aston Villa"/>
1.  ...
1.  <a class="d a" id="615278" title="Arsenal 1-1 Manchester United"/>']
2.  ,[6531,15,'Chelsea',3,38,22,9,7,75,39,36,75,4,19,12,5,2,41,16,25,41,3,
2.  19,10,4,5,34,23,11,34,
2.  '<a class="d a" id="615269" title="Liverpool 2-2 Chelsea"/>
2.  <a class="w h" id="615280" title="Chelsea 2-0 Swansea"/>
2.  ...
2.  <a class="w a" id="615298" title="Manchester United 0-1 Chelsea"/>
...

20.  ,[6531,171,'Queens Park Rangers',20,38,4,13,21,30,60,-30,25,20,19,2,8,
20.  9,13,28,-15,14,19,19,2,5,12,17,32,-15,11,'
20.  <a class="l a" id="615261" title="Everton 2-0 Queens Park Rangers"/>
20.  <a class="l a" id="615241" title="Fulham 3-2 Queens Park Rangers"/>
20.  <a class="l a" id="615261" title="Everton 2-0 Queens Park Rangers"/>
20.  ...
20.  <a class="d a" id="615285" title="Reading 0-0 Queens Park Rangers"/>
20.  <a class="l a" id="614130" title="Liverpool 1-0 Queens Park Rangers"/>']
```

C.1.2 legacy-matches.csv

```
1.  32,'Manchester United',614052,614115,614127,614161,614189,614265,
1.  614975,614983,614990,615006,615016,615096,615108,615111,615135,
1.  615140,615153,615044,615047,615082,615030,615068,615064,615159,615170,
1.  615180,615194,615207,615228,615245,615249,615266,615224,615270
...
20.  171,'Queens Park Rangers',614056,614116,614125,614168,614203,614307,614978,
20.  614986,614994,615008,615023,615094,615112,615119,615124,615142,
20.  615155,615031,615065,615086,615044,615050,615079,615161,615173,615187,
20.  615194,615208,615221,615225,615241,615252,615261,615272
```

C.1.3 fixture-info.csv

```
1.  MatchID,HomeID,AwayID,HomeTeam,AwayTeam,MatchDate,Result,
1.  'home_blocked_scoring_att','home_shot_off_target','home_ontarget_scoring_att',
1.  'home_post_scoring_att','home_total_scoring_att','home_won_contest',
1.  'home_fk_foul_lost','home_total_throws','home_total_tackle','home_total_offside',
1.  'home_won_corners','home_total_pass','home_accurate_pass','home_possession_percentage'
1.  'home_aerial_lost','home_aerial_won','home_goals','away_blocked_scoring_att',
1.  'away_shot_off_target','away_ontarget_scoring_att','away_post_scoring_att',
1.  'away_total_scoring_att','away_won_contest','away_fk_foul_lost',
1.  'away_total_throws','away_total_tackle','away_total_offside',
1.  'away_won_corners','away_total_pass','away_accurate_pass',
1.  'away_possession_percentage','away_aerial_lost','away_aerial_won','away_goals'
2.  614052,31,32,'Everton','Manchester United','08/20/2012 20:00:00',
2.  '1 :  0',2,9,7,2,18,7,13,18,19,0,6,275,196,30,18,28,1,4,6,4,0,14,7,11,
2.  23,15,1,8,646,571,69,28,18,0
3.  614115,32,170,'Manchester United','Fulham','08/25/2012 15:00:00',
3.  '3 :  2',4,9,7,1,20,8,12,20,25,1,8,614,571,59,5,5,3,7,3,6,0,16,12,7,
3.  12,24,0,8,409,366,40,5,5,2
4.  614127,18,32,'Southampton','Manchester United','09/02/2012 16:00:00',
4.  '2 :  3',4,7,4,0,15,3,8,19,21,2,4,487,402,45,11,7,2,4,9,7,1,20,4,4,17,16,
4.  2,7,587,518,54,7,11,3
```

C.1.4 upcoming-fixtures.csv

```
614129,15,31,’Chelsea’,’Everton’,’19/5/13’
614130,26,171,’Liverpool’,’Queens Park Rangers’,’19/5/13’
614133,23,13,’Newcastle United’,’Arsenal’,’19/5/13’
614134,18,96,’Southampton’,’Stoke’,’19/5/13’
614135,259,170,’Swansea’,’Fulham’,’19/5/13’
614136,30,16,’Tottenham’,’Sunderland’,’19/5/13’
614137,175,32,’West Bromwich Albion’,’Manchester United’,’19/5/13’
614138,29,94,’West Ham’,’Reading’,’19/5/13’
614139,194,24,’Wigan,’Aston Villa’,’19/5/13’
614132,167,168,’Manchester City’,’Norwich’,’19/5/13’
```

C.1.5 odds.csv

```
1,Chelsea,Everton,5Dimes,1.74,Pinnacle Sports,4.06,Interwetten,5.60
1,Liverpool,QPR,5Dimes,1.26,BetVictor,7.00,Pinnacle Sports,14.30
1,Newcastle Utd,Arsenal,Interwetten,6.50,Pinnacle Sports,4.52,Canbet,1.63
1,Southampton,Stoke City,Interwetten,1.80,Pinnacle Sports,3.93,Pinnacle Sports,6.21
1,Swansea,Fulham,Paddy Power,1.83,bet365,3.90,5Dimes,5.17
1,Tottenham,Sunderland,888sport,1.30,Pinnacle Sports,6.30,Pinnacle Sports,14.67
1,West Brom,Manchester United,888sport,4.33,Stan James,3.80,12BET,1.97
1,West Ham,Reading,Interwetten,1.70,5Dimes,4.20,BetVictor,6.50
1,Wigan,Aston Villa,888sport,2.10,5Dimes,3.74,bet365,4.00
1,Manchester City,Norwich,Canbet,1.32,Pinnacle Sports,6.21,Pinnacle Sports,13.94
```

C.2 Internet crawler

This section documents what the most important functions for the internet crawler do. Each PHP-script is documented in its own section.

C.2.1 whoscored_league_table_and_match_list.php

`updateRawTableInfo()` - the function to run for updating the *rawTable.csv* file.

`getWhoScoredTable()` - this function retrieves the table where all the match-IDs are stored at *whoScored.com.*

`updateMatches()` - the function for updating *matches.csv.*
`formatMatches($text)` - helper function that is given as argument a line from *rawTable.csv* and returns a string containing team-name, team-ID and all match-IDs found in that line,

separated by commas.

`getTeamNameAndID($text)` - helper function used by `formatMatches()` for finding team name and ID in a string-argument.

`updateLegacyMatches()` - updates *legacy-matches.csv*. Compares the new matches in *matches.csv* with the old ones in *legacy-matches.csv*, and for each that does not exist in *legacy-matches.csv*, adds it to the data-file.

`appendNewMatchesToList($legacy, $new)` - A helper function used in `updateLegacyMatches()`.Each parameter is a 2-dimensional array, each row should contain values such as "13, man utd, 12410, 1230141, 123102". This function adds any match-ID found in $new that isn't already in $legacy, and returns $legacy.

`updateAll()` - Runs all of `updateRawTableInfo()` , `updatematches()` and `updateLegacyMatches()`.

C.2.2 whoscored_match_ifo.php

`getNewMatchObjects()` - reads matchIds from *legacy-matches.csv* and retreives fixture-info for each match.

Crawling many web.pages takes some time, and we wish to be sure that the browser does not time out before we are done crawling. We therefore check for each match-ID whether it already exists in *fixture-info.csv*, and skip it if it does. We also have a timer `$timer` indicating how many seconds we may run before we stop the script, even though it is not finished.

`takenTooLongTime($startTime, $maxTime)` - returns *true* if we have surpassed the allowed processing time *$maxTime*, otherwise *false*.

`fixtureInfoAlreadyObtained($matchID)` - helper function for checking whether a match already has been found by checking for it in *fixture-info.csv*. Returns *true* if it has, otherwise *false*.

`getWhoScoredMatchInfo($matchID)` - helper function used by `getNewMatchObjects()`. is given a matchID and uses it to collect match info from *www.whoScored.com*.

C.2.3 get_upcoming_fixtures.php

`getNextMatches()` - this function retrieves all upcoming matches from *www.whoScored.com*, and overwrites *upcoming-fixtures.csv*.

`formatMatch($match)` Receives as argument a string value which is a line taken from the source code of *www.whoScored.com*. returns a string which gives the line in the correct format according to that of *upcoming-matches.csv*.

`getDefaultDateFormat($string)` - Receives as argument a date in string format, and transforms it from the form 'feb 23 2013' to '23/02/13'. This is for easier use in MATLAB.

`updateUpcomingMatchesFile()` - Uses the global variable *$nextMatchList*, which is an array containing all the next matches as strings, and overwrites *upcoming-matches.csv* with these lines. We do not append them, as matches already there may now have been played for all we know, making them no longer 'upcoming'. Therefore, we overwrite instead.

C.2.4 get_odds.php

`getBestOdds()` - Retrieves odds and names of the bookies that present the best odds at *www.betExplorer.com*, and updates *odds.csv*.

`findUpcomingMatchesStartPos($matches)` - The function is given as argument an array of match-IDs that represent the upcoming matches we wish to find/update odds for. It then searches through *odds.csv* until it finds the first occurence of one of the matches in the array, and returns its position. If none are found, the amount of lines (number of matches) in *odds.csv* is returned.

`getLinkToMatch($pair, $subject)` - Receives as arguments an array *$pair* that contains the two names of the teams participating in a specific match, and a string *$subject* containing the source code of the general page at the *www.betExplorer.com* domain for the the league we are interested in. Returns as a string value the URL for the page showing all bookies with odds available for the match we are looking at.

`addNewOddsToFile($count, $start)` Receives as arguments two int values; `$count` gives the number of matches we managed to get new odds for before the process had to be shut down because of time limits; `$start` gives the position in odds of the first match we have updated odds for.

This function overwrites the *odds.csv* file, leaving old match odds as they were, and either adding odds for new upcoming matches or updating odds for upcoming matches that we have obtained odds for previously.

C.2.5 simple_html_dom.php

`load_file($target_url)` - Is given as argument a string containing an URL value, which it uses to retrieve the source code of the web-page identified by the URL. We must first construct a `simple_html_dom`-object before using this function. An example use would be:

```
$target_url ="http://whoscored.com";
$html = new simple_html_dom();
$html->load_file($target_url);
```

C.3 Betting Simulator

In this section we present the most important methods for each MATLAB object/script in the simulator.

C.3.1 Game

`PresentGame(varargin)` - Is a simple method that presents the *Game*-object by showing which round it is taken from, which teams are playing each other and either *1)* how many goals were scored by each team or *2)* how the probability distribution is between the three outcomes if the game has not yet been played. `Varargin` Could contain a variable `verbose` which determines how much text is printed.

C.3.2 GameList

`GetGame(idx)` - Retrieves a *Game*-object from the list by using an integer argument *idx*, which gives the position of the match in the list.

`GetNoGames()` - Returns an integer giving the total number of games in the *GameList*-object.

`PresentGames(varargin)` loops through each *Game*-object in the *GameList*, and uses the *PresentGame()* function for each game. *Varargin* contains the variable *verbose*, which is passed on to the *PresentGame()* function.

C.3.3 Database

`LoadDataset(league, season)` - Uses two string parameters to identify which dataset is desired, and uses this to load the appropriate `.mat` file. This file will have been constructed using the *readData.m* script. The values from the *.mat* file will then be cached in the parameters given in table 3.2.

`HideResult(varargin)` - We may wish to predict the probabilities for results for a game that has already been played. `HideResult(varargin)` hides the amount of goals scored by each team for the desired games. *Varargin* give which matches to be 'blanked', either the game for a specific team in a specific round, the matches from a specific round, or matches from several rounds.

`GetIntensity()` - Calculates the intensity values for each team in each round, as given by the formula presented in Section 5.2.2.

`GetDominance()` - Calculates the dominance values for each team in each round, as given by the formula presented in Section 5.2.2.

C.3.4 Bookie

`ArrangeBets(db, punter, varargin)` - Arranges bets for a single round of matches. the parameter db is a *Database*-object which is used to pass on to *punter*, a *Simulator*-object. We use the *Simulator*-function *PredictGame()* to establish the probabilities for the three outcomes. *Varargin* contains information on which betting-strategy we will use. `ArrangeBets()` then determines whether the strategy and simulation-model gives positive or negative returns.

`SeasonSimulation(varargin)` - Given a starting round, runs `ArrangeBets()`for that round and each subsequent until the end of the season. *Varargin* contains, in addition to those used for `ArrangeBets()` , also variables for determining which JAGS-model to be used, and all the parameters needed for starting to run a JAGS-simulation.

C.3.5 Simulator

`GenerateSamples(db, varargin)` - the interfacing function that starts JAGS. JAGS fills the values in db that are not set yet, such as goals scored and other variables used depending on the model we are using (e.g. Maher, Dixon & Coles or ours). *Varargin* contains values for determining which model to use, how many chains and samples t use, how much thinning and how large burn-in. It also contains a *monitoring* variable giving all the values from the JAGS-model which we wish to be able to observe when JAGS is done.

`PredictGame()` - Asses the probabilities of the three outcomes for each match. This is done by, for each sample of goals scored, comparing goals for each opposing team in a match, and counting how many instances of home-win, away-win and draws that occur, and then divide by the total amount of samples. If the result of a match is already known, each sample of goals scored will be the actual amount, and thus the probability distribution will always be of the form `[0 0 1]`, `[0 0 1]` or `[0 0 1]`.

`GenerateTable()` - Generates the final table based on average points deduced from each match. If the probabilities for a arbitrary match is `[.1 .3 .6]`, the average points won by the home and away teams for this match will be:

$$Avg.points_{home} = SUM([3\ 1\ 0]^T \bullet [.1\ .3\ .6]) = SUM([.3\ .3\ 0] = 0.6$$

$$Avg.points_{away} = SUM([0\ 1\ 3]^T \bullet [.1\ .3\ .6]) = SUM([0\ .3\ 1.8] = 2.1$$

`SaveSamples()` - Each time we have run JAGS, we check a .MAT file *MASTER.mat*, containing a list of objects holding run-parameters of JAGS runs done in the past. These parameters would be the name of the model used, burn-in and sample sizes, thinning frequency and chains used, and the name of a .MAT file containing the results of the said run. After each run of *GenerateSamples()*, *SaveSamples()* checks if this specific JAGS run already exists in *Master.mat*. If it does not, we add this runs parameters to that file, and create a new .mat-file containing all the results for the current run done.

`LoadSamples()` - Checks if the JAGS-run about to be executed has been done before; if yes, load that .mat file where results are held, otherwise run JAGS.

C.4 Collecting data from a league

This section describes how to alter which football league and season we are collecting data from.

Step 1: Change target URLs in `whoscored_league_table_and_match_list.php` and folder names for which .csv files to use. We must also alter the stageID value of the league, as this is used explicitly when matching regular expressions. The following global variables must be changed:

```
$target_url_for_whoscored_league_table =
  "http://www.whoscored.com/Regions/252/Tournaments/2
  /England-Premier-League";
$target_url_for_whoscored_upcoming_fixtures =
  "http://www.whoscored.com/Regions/252/Tournaments/2
  /Seasons/3389/Stages/6531
  /Fixtures/England-Premier-League-2012-2013";
$target_url_for_betexplorer_odds =
  "http://betexplorer.com/soccer/england/premier-league/";

$league_and_year= "EPL 2012-2013"; //Folder name for .csv files

$stageID = "6531";
```

Their current values have been included as examples of what the URLs should look like.

Step 2: In `translateForMatlab.php`, we must change the filename (the variable `$file`) of where we wish to save the translated .csv file which has been formatted to be correctly read by the Matlab script `readData.m`.

Step 3: Finally, in `readData.m` we must alter the location of the .csv file we wish to import data from and construct a matrix out of.

C.5 Using the Simple_html_dom.php Script

Below is an example of how to use the functions provided by *simple_html_dom.php* to download the source-code of a website. We may take note that any PHP-code will not be visible in the source-code, showing only HTML and javascript.

```
$html = new simple_html_dom();
$html->load_file($target_url_for_whoscored_league_table);

$subject = $html;
```

subject will then contain the content of the provided web-page's source code as a single string.

C.6 Setting up the WampServer

After having followed the installation guide at `http://www.wampserver.com/en/#download-wrapper`, the crawler-folder should be placed inside the *www* folder. The server should be offline at all times, as any external users of the crawler methods will not have full access to write and append data to the .csv files we are using.

If the location `localhost:80` can not be reached, it is probably because that port is being used by another application. The port used by the WampServer can be changed by accessing `C:/wamp/bin/apache/apache2.2.22/conf/httpd.conf`, altering the line:

`Listen 80`,

and choosing any 4 digit value instead of 80, which is not already in use.

The interface to the crawler methods can then be reached at `http://localhost:8081/internetcrawler/hp/premier-league-table.php`.