UPPSALA
UNIVERSITET

Statistical Football Modeling

A Study of Football Betting and Implementation of Statistical Algorithms in Premier League

Jonas Mirza, Niklas Fejes
Supervisor: David Sumpter

Project in Computational Science: Report

January 29, 2016

PROJECT REPORT

Department of Information Technology

Contents

1 Introduction

1.1 Making money on football betting?

This report covers our work on football betting and starts with a brief introduction to the field followed by some needed theory. Thereafter, the implementation of the three different models we have been studying will be covered, to finish with our results and discussion.

The idea for all the models is taken from the book Soccermatics[1] written by our supervisor David Sumpter.

The main goal in our work was to test and improve different betting methods and examine the possibility to make money in the long run. This was done by building mathematical models, based on statistics, which exactly tell how to place the money. All our models performed better than a random-betting model and one of the models was able to make money during the validation period.

1.2 Betting and bookmaking

There are plenty of different scenarios that one can bet on when it comes to sports. For any given Premier League game one can easily find over 100 different types of bets, everything from who will win to who will receive the first yellow card. However, in this project, only bets of the type "singles" in Premier League were analyzed. A single bet is a bet placed on just one selection. In football that yields win, draw or loss $(1, X, 2)$, from a home team point of view.

A typical single bet can look something like $(1.72, 3.80, 4.50)$ which means one have a chance to win 1.72 times the money if betting on home win and so on. So how do the bookmakers set the odds? If gambling had been a fair game the odds should correspond to the estimated probability for the outcome they represent. In this case home win will give 1.72 the money and therefore the probability for it would be its inverse 0.58. However, this is not the case and a simple example can show why. If one takes the inverse and sums up the probabilities for all the outcomes in one game one expects the sum to be equal to one, but for the bets stated above the sum is 1.07 which means there is a 7% margin added by the bookmakers. Further on, the bookmakers have no real interest in predicting the outcome themselves. On Wikipedia one can read:

> "*A bookmaker strives to accept bets on the outcome of an event in the right proportions so that he makes a profit regardless of which outcome prevails.*" [2]

[1] ISBN: 9781472924124

[2] `https://en.wikipedia.org/wiki/Mathematics_of_bookmaking` (25/1 2016)

This implies that the odds will be adjusted accordingly to the demand and that the given odds rather represent the public opinion about the outcome than the "true probability". In conclusion, if one could formulate a method that predicts the outcomes better the public opinion plus the bookmaker's margin one could make money from betting.

1.3 Problem description

The idea of this project is to analyze three different betting models and examine if it is possible to use them to make money in the long run. A big part of the project was also to create a live web application that presents our results and shows how the models perform in the current and future seasons of Premier League.

A necessary condition to make money in the long run is that the model's prediction p^* must give a better prediction of the outcome compared to the odds b. To have a chance to succeed with this task the models need to have a statistical foundation. Therefore, the analyzed problem can be formulated as follows: Create a model that uses historical football data and returns a set of prediction values, for $(1, X, 2)$, that predicts the outcome better than b.

2 Theory

2.1 Odds and probabilities

The odds we are using are given in a format known as "European style" in the gambling community, which for a fair (no-margin) bet is given as odds $= 1/\mathrm{P}(\text{win})$ as described in the introduction. While it is impossible to know exactly how the bookmakers set their odds, we should not assume that they are setting the odds by the best possible prediction of the match outcomes. Instead, they most likely weight together their own predictions; how much money they receive in bets for each outcome; and how other bookmakers have set their odds.

What we can do however, without confining ourselves, is to pretend that the odds represent those given by a naive bookmaker who has predicted the match outcomes to her best, set the odds as the reciprocal of the probability, and scaled them down by some percentage to take a revenue only on the winning bets. Formulating this model mathematically gives the motivation for computing the probabilities as

$$\begin{bmatrix} \mathrm{P}(\text{home}) \\ \mathrm{P}(\text{draw}) \\ \mathrm{P}(\text{away}) \end{bmatrix} = \begin{bmatrix} 1/\text{odds}_1 \\ 1/\text{odds}_X \\ 1/\text{odds}_2 \end{bmatrix} \cdot \frac{1}{\sum_{i \in \{1,X,2\}} 1/\text{odds}_i},$$

where the normalizing factor is needed in order to remove the margin from the odds. If the match results were to be distributed exactly by these probabilities, we would always lose in the long run due to the bookmaker's margin. This is easily seen in the following example:

> Let p be the probability of outcome A in a game with two outcomes (A or B). The odds b_A is, as previously described, set by the bookmaker as
>
> $$b_A = 1/p \cdot (1 - m) \tag{1}$$
>
> where m is the bookmaker's margin such that $0 < m < 1$. The expected net gain when betting x units is then
>
> $$\mathrm{E}[\text{“net gain”}] = p \cdot ((b_A - 1)x) - (1 - p) \cdot (-x) = (pb_A - 1)x.$$
>
> Inserting Equation 1 then gives us
>
> $$\mathrm{E}[\text{“net gain”}] = -mx.$$

In the example the same equation applies for the odds on outcome B, and in general to all odds on a game with any number of outcomes. This implies that however we place our bets the average net gain will tend towards $-mx$ where x is the total amount we bet, and we are expected to lose in the long run.

However, our assumption was that the bookmakers do not set their odds by the best possible predictions, so we should not assume that Equation 1 holds. If we can improve the estimates in any way we can make a net gain on the odds.

We can study these odds probabilities by looking at the historical odds that the bookmakers have set in the last years. Figure 1 shows the probabilities derived from the best available odds for the 2648 games played in Premier League between 2004 and 2015.

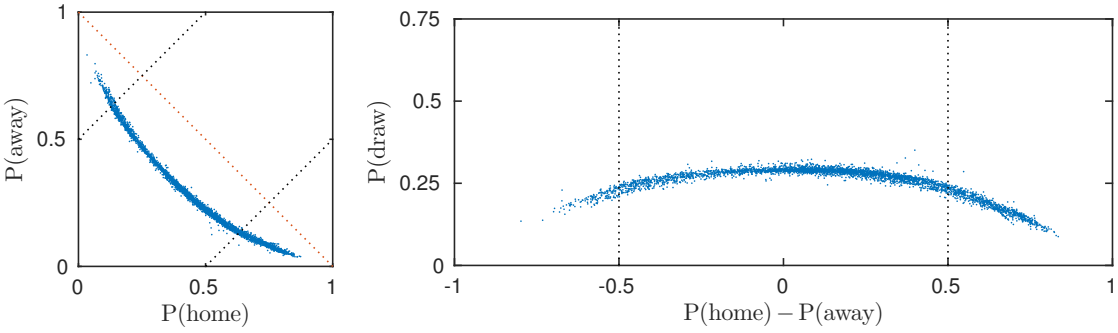


Figure 1: Scatter plot of the probabilities of home win versus away win in 2648 Premier League games.

It is clear that the bookmakers prefer to put their probabilities close to a fixed curve in the "home-away" plane, since the probability coordinates in theory could be placed anywhere in the triangle under the dotted diagonal line between $(0, 1)$ and $(1, 0)$ in the left plot.

In the left plot, the probability of draw P(draw) for a coordinate $(\text{P(home)}, \text{P(away)})$ can be seen as the shortest distance between the diagonal line where $\text{P(home)} + \text{P(away)} = 0$. This can be derived from the sum of the three possible outcomes, i.e.

$$\text{P(home)} + \text{P(draw)} + \text{P(away)} = 1.$$

The right plot in Figure 1 shows the probability for draw versus the difference between the probabilities for win for either team. The transform between the two projections is linear, such that the shape of the curve is the same in both plots. The projection in the right plot [P(home) − P(away) vs. P(draw)] is useful for visualizing how an odds-based model transform the probabilities, since the variable on the x-axis can be seen as a measure of the home team advantage on a scale from -1 to 1. Another projection one can use is to plot P(home win $\mid \neg$ draw) on the x-axis which yields a visually similar plot. In the rest of the report we have chosen to use the difference projection since it is slightly easier to use.

2.2 Poisson and Skellam distributions in football

A common way to model a football game is to assume that the expected number of goals a team will make is given by a Poisson distribution. This assumption aligns well with the actual match results, which can be seen from the histograms in Figure 2.

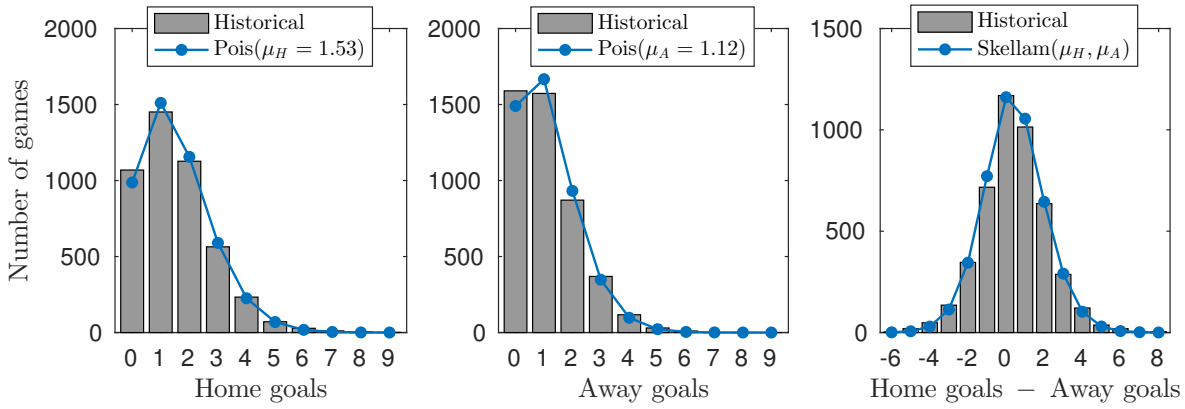


Figure 2: Plots showing that the distribution of the number of goals in the matches in 2003–2014 resembles a Poisson distribution.

The distribution of goals for the home team appears to have the same pattern, and by further assuming that the number of goals for the home team (X) and away team (Y)

are Poisson distributed, i.e.

$$X \sim \text{Pois}(\mu_1) \qquad \text{and} \qquad Y \sim \text{Pois}(\mu_2),$$

we get that the probability for home win and draw are

$$\text{P(home)} = \text{P}(X - Y > 0) \qquad \text{and} \qquad \text{P(draw)} = \text{P}(X - Y = 0).$$

The distribution of the difference between two independent random variables with Poisson distribution is known as the *Skellam* distribution, i.e.

$$X \sim \text{Pois}(\mu_1), \quad Y \sim \text{Pois}(\mu_2) \quad \Longrightarrow \quad (X - Y) \sim \text{Skellam}(\mu_1, \mu_2).$$

This distribution can be used in a very simple model for a game, where the number of goals for each team are modeled as independent variables with Poisson distribution, of expected value μ_1 and μ_2. Since $(X + Y) \sim \text{Pois}(\mu_1 + \mu_2)$, we can further constrain the model by assuming that the expected value of the total number of goals $\mu = \mu_1 + \mu_2$ is constant. In Figure 3 the contour lines where μ are constant are shown.

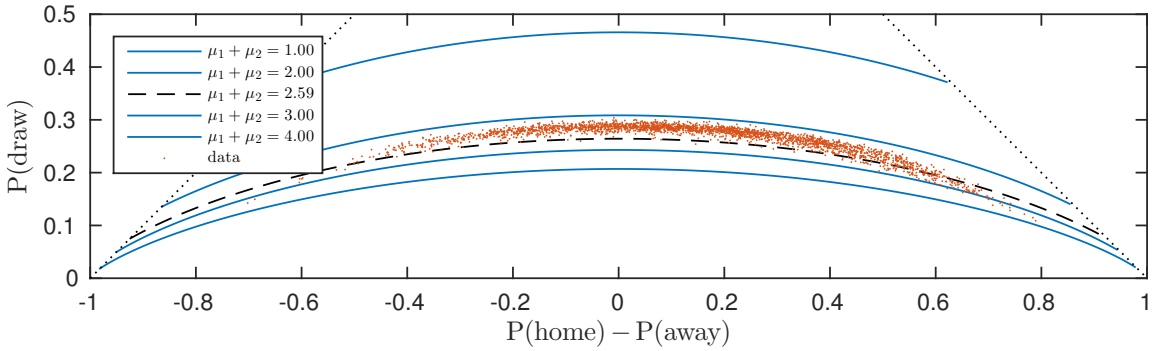


Figure 3: Blue lines have constant expected number of goals ($\mu_1 + \mu_2 = K$). At the dotted diagonal lines, the probability that one of the teams will make no goal is 1, and thus the probability of the other team winning is their probability of making at least one goal.

In Figure 3 we can see that the bookmakers set their odds approximately as the predictions of this model. The contour line for $\mu = 2.59$ is shown in the plot, which is the average number of goals per match over the last 10 years. The probability for draw is placed somewhat higher than a "pure" Poisson distributed match would suggest, which might depend on many other factors. For example a team with a low number of expected goals may play defensive against a better team and aim for a draw rather than a win. In Figure 2 the histograms suggest that the probability of one team making no goals is slightly higher than predicted by the Poisson distribution, which could be a reason to slightly raise the probability of draw.

2.3 The Kelly Criterion

There is a well-known formula in the betting community known as the Kelly Criterion, which is an equation for computing the optimal bet to place in order to maximize the expected outcome, given odds and outcome probabilities as input.

The Kelly Criterion is stated as follows when we have a two-outcome game where the results are A or B, and the variables are:

p_A, p_B – probabilities of either outcome, such that $p_A + p_B = 1$,

B_A, B_B – odds (European style format),

x_A, x_B – fraction of payroll to bet on A or B, given that you decide to bet on one (and only one) outcome.

The formula states that in order to maximize the long-term expected gain, the optimal fraction to bet is given as

$$x_A = \frac{p_A\, B_A - 1}{B_A - 1} \quad \text{and} \quad x_B = \frac{p_B\, B_B - 1}{B_B - 1}.$$

If $x_A \leq 0$ you should not bet on A, since you are not allowed to place a negative bet with the same odds.

The formula is easily derived by finding x_A that maximizes the expected value of the logarithm of the "net wealth when playing on A", W_A. This value is the multiplicative change in size of the payroll, and for a game played on outcome A it is defined as

$$W_A = \begin{cases} 1 - x_A + B_A\, x_A & \text{if win,} \\ 1 - x_A & \text{otherwise.} \end{cases}$$

The Kelly Criterion then gives the solution to the maximization problem

$$\underset{x_A}{\operatorname{argmax}}\ \mathrm{E}[\log(\mathrm{W_A})] \;=\; \underset{x_A}{\operatorname{argmax}}\ p_A\, \log(1 - x_A + B_A\, x_A) + p_B\, \log(1 - x_A). \qquad (2)$$

The reason for optimizing this expected value can be derived from expected utility theory[3] which uses the idea that the value of the gain should not be proportional to the monetary value, but rather the utility of the money. In short, one should in each game try to maximize the *utility* (in this case the logarithm) of one's wealth instead of the wealth itself. For example if your wealth is \$100; losing \$99 is far worse for your long-time betting compared to the gain you get by winning the same amount. In this case a 51% chance of winning would make the expected value of W_A (without the utility) larger than 1, so the statistics tells you to bet as much as you can. If one instead uses the expected value of the utility, the equations will balance out the risks over the gains in a way that is more sensible in the long run.

[3] `http://plato.stanford.edu/entries/rationality-normative-utility/`

An advantage of using the logarithm as the utility function is that if you have a finite probability for outcome B (i.e. $p_B > 0$), it is never optimal, regardless of the odds, to place everything on A since this, for outcome B, would make the utility value $\log(0) = -\infty$. This corresponds to never taking the risk of losing all your money.

The formula can also be motivated as a probability-weighted geometric product of W_A,

$$G(x_A) = (1 - x_A + B_A\, x_A)^{p_A}\, (1 - x_A)^{p_B}, \tag{3}$$

since a consecutive bet would be multiplied by $G(x_A)$, i.e. if you bet N times you would expect to have a total gain of

$$G = \prod_{k=1}^{N} G_k(x_{A,k}).$$

Three outcome bets

The problem gets more complicated when we instead have three different outcomes such as a football game. We now have the outcomes *win*, *draw* or *loss* for the home team, which we label 1, 2 and 3. The variables are then:

p_1, p_2, p_3 – probabilities of either outcome, such that $\sum_i p_i = 1$,

B_1, B_2, B_3 – odds,

x_1, x_2, x_3 – fraction of payroll to bet on each outcome.

Since we in theory are allowed to bet on multiple outcomes, we should find the three values $\mathbf{x} = (x_1, x_2, x_3)$ which maximize the expected outcome given the variables above.

The net gain, given that outcome i happens is

$$G(x_1, x_2, x_3; i) = \begin{cases} 1 - \sum_j x_j + B_1\, x_1, & \text{if} \quad i = 1 \\ 1 - \sum_j x_j + B_2\, x_2, & \text{if} \quad i = 2 \\ 1 - \sum_j x_j + B_3\, x_3, & \text{if} \quad i = 3, \end{cases} \tag{4}$$

and the expected gain, with the same reasoning as in the binary outcome case, becomes

$$G(x_1, x_2, x_3) = \prod_{i=1}^{3} \left(1 - \textstyle\sum_j x_j + B_i\, x_i\right)^{p_i} \tag{5}$$

or

$$G(x_1, x_2, x_3) = \exp\left(\sum_{i=1}^{3} p_i\, \log\left(1 - \textstyle\sum_j x_j + B_i\, x_i\right)\right). \tag{6}$$

If we limit ourselves to only one bet ($x_1, x_2,$ or x_3), which should be positive, we find that the optimal bet is given by the same equations as in the binary case

$$x_i = \frac{p_i\, B_i - 1}{B_i - 1}. \tag{7}$$

If multiple x_i's are positive, we should choose the one which has the highest gain $G(x_1, x_2, x_3)$. It should be noted that $G(x_1, 0, 0) > G(0, x_2, 0)$ does not imply that $x_1 > x_2$; and if we have two positive x_i's the optimal combined bet is not given by the formula above.

We thus have a formula for computing the optimal fraction of one's bankroll to place on a 1X2-bet given that we have an estimate of the true probabilities of each outcome and the odds received on the wager. The odds B_i are given by the bookmakers, so what remains is to find good approximations of p_i. In the rest of the report we drop the index, and denote the set of these three probabilities as p^*.

The Kelly Criterion is useful for evaluating the long-term performance of a betting model, and in the project we have been using it in order to evaluate the models we have implemented.

2.4 Multinomial Logistic Regression

In order to use the odds to predict the actual outcomes, which we do in the Odds Bias model (Section 3.3), we need to use some method to fit the odds to the results. This problem is non-trivial in several ways, specifically that we have three discrete outcomes in a game, that the odds themselves are not good linear predictors, and that we need to specify a model that cannot be too general since the number of games in the last seasons are few from a computational perspective.

We have investigated a couple of different algorithms, and the most prominent has been multinomial logistic regression (MNR) implemented by MATLAB's `mnrfit`. The alternatives to this method are the generalized linear models present in MATLAB (`glmfit`, `stepwiseglm`, etc.) but since these only take binary outcomes we decided to use MNR instead of weighting together three binary models. MNR is similar to binomial logistic regression, which is the standard way to fit a predictor to a binary outcome, but instead uses a multinomial distribution.

In order to make a fit we need a model. In short, it should take the odds for home win, draw and away win, and output the three probabilities for each outcome. While we could simply feed the odds directly into the MNR, this would not be able to make good predictions since we do not expect the odds to be linearly dependent on the probability of the outcomes. We therefore should preprocess the odds in some way to get a model that is valid through dimensional analysis. Figure 4 shows the basic flowchart for the model.

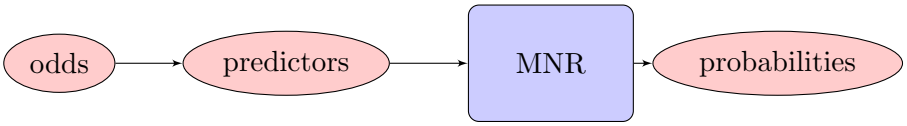


Figure 4: Flowchart for the model.

To convert the odds to predictors we take the following steps:

1. **odds:** The odds are taken directly from the source, see Section 3.5.
2. **odds probability:** The odds are converted to probabilities, by the formula assuming that the bookmakers odds are based on a fair game. (See Section 2.1.)
3. **predicting probabilities:** In this step we choose which odds probabilities to use for the prediction. In addition to the three outcome probabilities, we found that adding P(favorite win) as a predicting variable improves the results significantly.
4. **logit predictors:** In the final step before the MNR we transform the probabilities by the logit function, defined as $\text{logit}(p) = \log(p/(1-p))$. (See motivation below.)

In a (binomial) logistic regression, we are essentially doing a linear fit of a predictor to the *logit* of the outcome probabilities. Since the predictors are also probabilities, it is reasonable to apply the logit function to them before we use them, giving us the following model in the binary case.

$$\text{logit}(p^*) = \beta_0 + \beta_1 \text{logit}(p).$$

This model has the benefit that the probabilities are always in the range $(0, 1)$, and it thus weights the variable in a more sensible way than just using the probabilities. For example, input probabilities of 0 or 1 will map to $\pm\infty$ (in the linear model space), and the model will never be able to output unreasonable values such as -0.1 or 1.1. In the multinomial case the *softmax* function is used instead of the logistic function, and the equations are changed accordingly.

The motivation for using P(favorite win) as predicting variable comes from that we, by inspection, think that the peak in the draw probability curve (Figure 6 (a)) is lower than it should be and that it not necessarily must be smooth for even matches. The favorite win probability, defined as

$$\text{P(favorite win)} = \max(\text{P(home win)}, \text{P(away win)})$$

introduces the non-linearity seen in Figure 6 (b), and allows the MNR fit to model a non-smooth peak.

In short, the full model is then described by the following equations. Let the vector containing the predictors $(x_1, x_2, x_3, \dots)$ be

$$X = \begin{bmatrix} 1 & x_1 & x_2 & x_3 & \cdots \end{bmatrix}^T,$$

and the training parameters be

$$\beta = \begin{bmatrix} \beta_0^1 & \beta_1^1 & \beta_2^1 & \beta_3^1 & \cdots \\ \beta_0^2 & \beta_1^2 & \beta_2^2 & \beta_3^2 & \cdots \end{bmatrix}.$$

Then the probabilities p are computed as

$$p = \begin{bmatrix} \exp(\eta_1) \\ \exp(\eta_2) \\ 1 \end{bmatrix} \cdot \frac{1}{\exp(\eta_1) + \exp(\eta_2) + 1}, \qquad \text{where} \qquad \eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \beta\, X.$$

It should be noted that this equation is essentially the softmax function[4].

MNR will attempt to find the parameters β such that the model predicts the training data as good as possible, which is all done in Matlab's `mnrfit`. There are further several training parameters that can be tuned, which are discussed in MATLAB's manual.

3 Methods

This section covers the implementation of the three models that has been studied in this report, namely the Expected Goals model, the Elo model, and the Odds Bias model. A Common feature for all the three models is that they return a set of p^*. By the help of the Kelly criterion the models can then be evaluated and the fraction of one's payroll to bet can be established.

3.1 Expected Goals

Idea

A rather popular model type in the football world is the Expected Goals model. The model attempts to predict how many goals each team will make during their encounter and estimate p^* based on that. The model uses "shots on goal" as input data and returns the expected number of goals for a team. And by the use of the Skellam distribution p^* will be obtained.

[4] `https://en.wikipedia.org/wiki/Softmax_function`

The reason why this model type gained ground is that models that uses "goals" as input data needs a very large data set in order to have a low variance. A large data set implies that the model takes matches from a wide time span and will therefore not be trend secretive. If one instead uses "shots on goal" an equal large data set can be obtained in just a few matches since there are far more shots on goals compared to goals. Models that are based upon expected goals can therefore be expected to be more trend sensitive compared to goal models. In this case trend sensitivity refers to how long time it takes for a model to adjust when a dramatic change happens in the league. A typical dramatic change can be; a key player quits, a new coach etc.

A crucial part of this model is to be able to transform "shots on goals" into "goals" and this was done in the following fashion. Old data that showed from where a shot had been kicked and whether it was a goal or not were gathered. A specific function was then fitted to the data in such way that the function had the shot positions as input and returned the probability of making a score from that very position (p_g).

If one now is interested in predicting a specific encounter one simply takes the shot data for a few matches back in time for both of the team, plug every shot into the fitted function and sum up the probabilities, and one ends up with the expected number of goals for both the teams. This numbers can then be plugged in to the Skellam distribution which will return p^*.

Implementation

All shot data for Premier League were loaded into MATLAB. The football pitch was then divided into an 40×40 equal sized rectangular grid. A 3D histogram that showed the relative goal frequency for every rectangle was created, see Figure 5.

It was noticed that the graph in Figure 5 reminded of a function that describes the goal angle (θ) from a certain location (x_k). Let $\bar{u}$ and $\bar{v}$ denote the vectors from x_k to the goal posts. Then the goal angle can be defined as $\theta = \frac{\bar{u} \cdot \bar{v}}{|\bar{u}||\bar{v}|}$. A constant c was then optimized in such way that $c \cdot \theta$ gave the best fit to the 3D histogram. p_g was then computed as $p_g = c \cdot \theta$.

Shots from the validation year were then loaded in MATLAB and a few games back in time were used as input to p_g and summed up to determine the Expected Goal for both teams in a given encounter. This number was fed into the Skellam distribution and p^* was obtained.

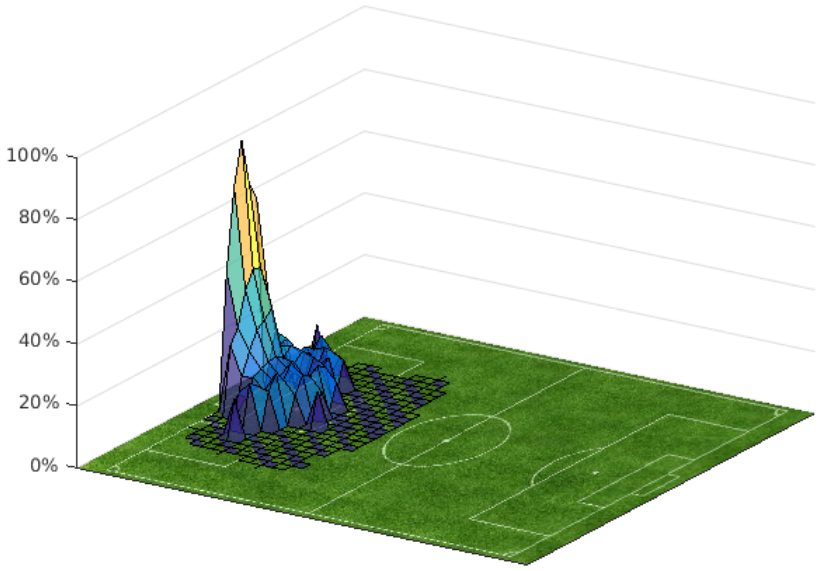


Figure 5: 3D histogram of the probability to score from a given box.

3.2 Elo model

Idea

A common rating system in all types of sport is named after its creator Arpad Elo[5]. The system was first developed to rate chess players but has evolved through the years and is now used to rate everything from football to e-sports.

The idea is that every player/team has an Elo-rating, a number, corresponding to their skill level. If a team with a high score plays against a team with a lower scorer, the better team i.e. the team with a higher score, is expected to win and will thus not increase their rating by much if they win. The team with the lower score however will not be expected to win and will be awarded a lot of points in the event they do. In this fashion the Elo-table will get updated. The Elo-table can then be used to predict the outcome of a given encounter.

Since the Elo-model was developed for chess, which is a game with binary outcomes, we made some tweaks in order to apply it on Football. This was done by first applying the binary Elo-model to the encounter and then use old data to estimate the probability for draw. Finally half the probability for draw was subtracted from both the probability of win and lose so that the whole set sums to one.

[5] `https://en.wikipedia.org/wiki/Elo_rating_system`

Implementation

All match data for Premier League were loaded into MATLAB where all the teams in the league were designated the same Elo-score. The Elo-table was then updated by the use of all the match result data in the time interval, in the following way. Let T_1 and T_2 denote the teams old Elo-rating and T_{n1} and T_{n2} denote their new Elo-rating after a given encounter.

If then T_1 wins the encounter, their new Elo will be

$$T_{n1} = T_1 + 40\left(1 - \frac{1}{1 + 10^{\frac{T_2 - T_1}{400}}}\right)$$

and T_{n2} will be

$$T_{n2} = T_1 + T_2 - T_{n1},$$

and vice versa if T_2 wins.

If the match was a draw

$$T_{n1} = T_1 + 40\left(\frac{1}{2} - \frac{1}{1 + 10^{\frac{T_2 - T_1}{400}}}\right)$$

and vice versa for T_2.

With the updated Elo-table the probability for win, loss (W, L), for T_1, can be calculated as:

$$W = 1 - \frac{1}{1 + 10^{\frac{T_2 - T_1}{400}}}$$

since this is the very same distribution used to score the teams with in the first place, and then $L = 1 - W$, this given a zero probability for draw.

To get the probabilities for win, draw, and loss (p_1^*, p_X^*, p_2^*) one have to adjust for the draw probability. This was done by fitting historical outcomes to historical odds by the use of MNR. The odds were then interpreted as probabilities and P(draw) was plotted against P(home win $\mid \neg$ draw). This plot was then used to get p_X^* as a function of W and L. p_1^* and p_2^* was then given by subtracting $\frac{p_X^*}{2}$ from W and L.

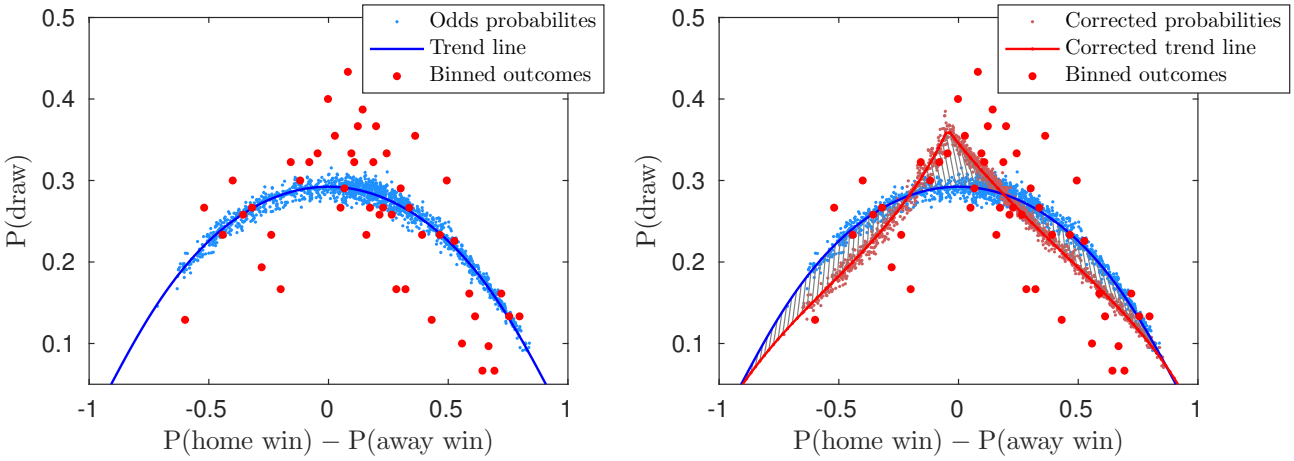
3.3 Odds Bias

Idea

In the other two models we have used rankings and expected goals in order to get p^* but what if we use the odds to approximate p? We can easily get a set of probabilities from the odds as described in Section 2.1, but directly applying the Kelly Criterion to these will, due to the bookmaker's margin, result in that no bet is expected to be profitable.

Section 2.1 also shows the distribution of these probabilities, and if we add the actual match outcomes after binning them, we get the plot shown in Figure 6 (a). To implement the binning, we sort the odds by P(home) − P(away), divide them into equally sized bins, and compute the average number of draw games in each bin. This will give a good visual representation of the match outcomes, and it should approximate the discrete outcomes used in the MNR.

The idea behind the Odds Bias model is that in the last few years, the trend line in the odds has not been well-aligned with the actual outcome of the matches. In Figure 6 this can be seen from that many of the bins for even games, where P(home) − P(away) ≈ 0, are above the odds trend line.



(a) Trend and odds. (b) Model fit.

Figure 6: Visualization of the odds. The odds used are the maximum of the four biggest bookmakers from 2011–2014, and each bin represents the average 30 games.

Implementation

The theory and mathematical model is described in Section 2.4. In short, we perform a regression (MNR) based on the historical odds for a few years back. We decided to use the seasons 2011–2014, since this gives us 1520 matches to train our model on. Using more seasons would make the fit less sensitive to noise, but on the contrary it would not be able to catch the trends that have appeared in the last few years.

Figure 6 (b) shows how the trend line and the probabilities are moved after the fit. The non-smooth peak in the middle of the fit is due to a non-linearity we introduced in order to remove the assumption that the curve should be smooth when P(home)−P(away) = 0, see Section 2.4 for details. While this may seem to contradict the Skellam distribution based model we discussed in Section 2.2, we are in this model trying to make a fit to the actual data, and not the theoretical model.

The model was trained with MATLAB's `mnrfit` implementation, and the fit has shown promising results when used in the 2015/2016 season.

3.4 Random betting model

In order to analyze the models in the Sections 3.1 to 3.3 a random betting model was developed. The model was created in order to have an information free model which implies that anything that performed better than this model contained some sort of information.

The random betting model randomly picked a a set of p^* from a distribution created from the odds curve, the curve in Figure 1.

3.5 Data and data sources

In order to collect data about the matches we have been using several different sources. The sources we have been using for the historical odds are from `football-data.co.uk`[6] which provides weekly updated csv files with historical odds given by many different bookmakers, as well as maximum and average odds.

This data also contains information such as number of shots on goal, number of corners, half-time and full-time scores, number of fouls, and number of red and yellow cards.

There are some problems with this data though, and that is that different bookmakers have different ways of earning money. Just looking at the highest odds available at any of the over 20 sites that one can play at might be misleading, since some bookmakers require that you to pay a certain percentage when you deposit your money, and some might request you to place a certain amount to be able to get the best odds. Therefore, for our training, we have only been using the maximum odds by the four biggest bookmakers. This should ideally reduce the noise in the data, caused by odds adjusted for local marketing offers.

We have been using `clubelo.com`[7] to get historical Elo-ratings in addition to the ones we have computed ourselves from the match results. This site provides an API for retrieving Elo-ratings for all Premier League teams and games, as well as for most other leagues. While there are other team rankings to be found on the web, this one was the easiest to use and the difference between Elo-rankings from different sources should be small since the equations are the same.

[6] `http://football-data.co.uk/englandm.php`
[7] `http://clubelo.com`

4 Results

This section starts with evaluations for all the models during the same time span, this in order to be able to compare them. Next, we continue with further evaluations of the Odds Bias model to end with a short presentation of the web application that was built. All plots in this section shows the gain, in percent, as a function of number of games. This implies that if a validation is above the zero-line it has gained money, and if it is below it is losing money. Due to the nature of the Kelly criterion we will never be able to lose more than 100%, which is why some of the plots are presented with the y-axis in log scale.

4.1 Model comparison

The models were set up with a calibration interval from 2007–2010 and then validated from 2011–2014. The odds that were used were the best odds from `footballdata.co.uk`. Figure 7 shows these validations, by the use of the Kelly criterion, in semi-logarithmic plots.

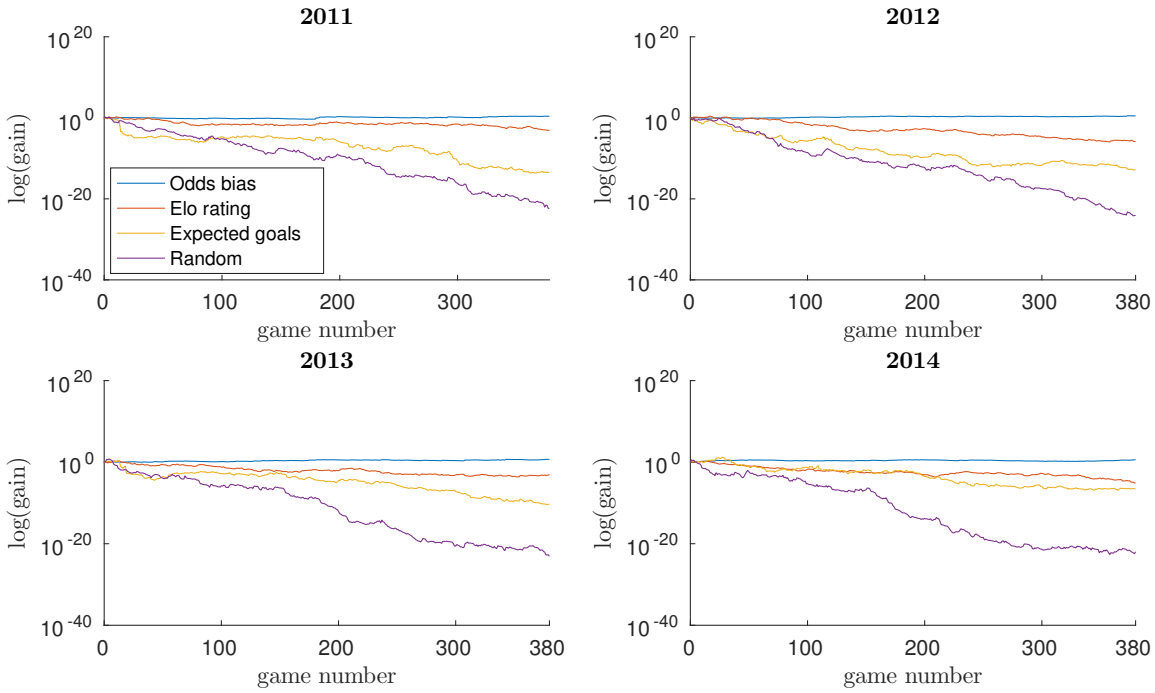


Figure 7: Four semi-logarithmic evaluations, by the use of the Kelly criterion, of how well the models would have performed from 2011–2014.

In Figure 7 it can be noticed that the Elo- and Expected Goals- models evaluation, after a few steps, lies between the zero line and the evaluation of the random model. Further on the validation of the Odds Bias model lies on or above the zero line.

4.2 Odds Bias

Since the Odds Bias model evaluation is over the zero line in figure 7 the same evaluations, for 2012–2015, is presented in Figure 8 but now with linear axis.

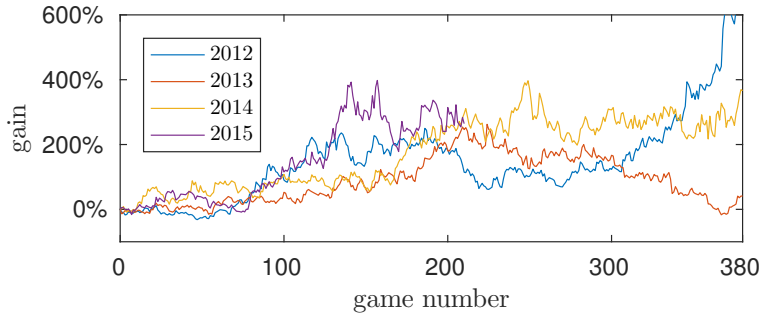


Figure 8: Linear evaluation, by the use of the Kelly criterion, of the Odds Bias models performance from 2012 to 2015.

Note that the evaluations is above the zero line for all years except of 2013.

As expected the evaluation differed when the predictors in the regression were changed. Figure 9 shows the evaluation of two Odds Bias models with different predictors. It should be noted that the choice of predictors is not trivial since too many predictors will lead to overfitting, and using linearly dependent predictors will make the MNR fit misbehave. The calibration interval was 2011–2014 and they were validated for 2015.

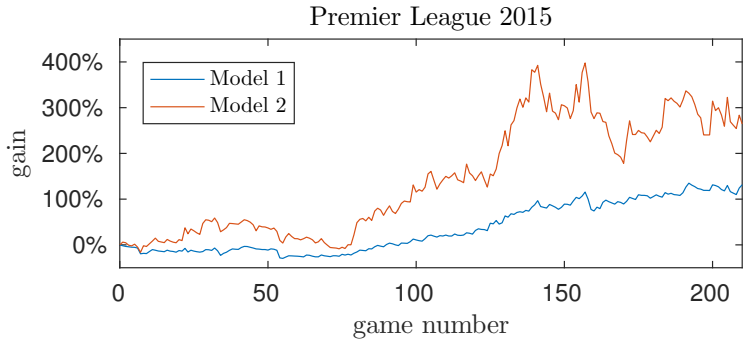


Figure 9: Linear evaluation, by the use of the Kelly criterion, of the Odds Bias models performance during 2015. The blue line corresponds to using (win, draw, lose) as predictors, and the red line corresponds to (win, draw, lose, favorite).

As can be seen in Figure 9 the two evaluations differ. The blue line is smoother than the red one but do not rise as high. The smoothness in Model 1 can be explained by the fact that the predicted probabilities are closer to the odds probabilities, so the bets will in general be smaller. The increased profit in Model 2 comes mostly from the even matches, which indicates that those games are modeled better than in Model 1.

4.3 Web application

The web application that was built as a part of this project can be found at: `www.betamatics.com`. It contains an evaluation of the Odds Bias model that updates every day, as well as a table over which bets to place for the next week. There is also a "Strategies" section that briefly explains how the Odds Bias- as well as the Elo- and Expected Goal models work, and one can also read more about bookmaking and odds.

5 Conclusions

The results in Figure 7 show that the Elo- and the Expected Goals model performed better than a random-betting strategy, but were far from the zero line. We can therefore conclude that both these approaches give some information but not enough to make money on their own.

The Odds Bias model had a clear upward trend for all the validation years except for 2013 as shown in Figure 8. We can also conclude that the choice of predictors in the regression is of great importance as seen in Figure 9, and in it self is a whole subject for future studies.

6 Discussion

6.1 Problems

Selection of bookmakers

The data we used came from `footballdata.co.uk` that keeps track of roughly 20 bookmakers. In the earlier stages of our work we used the best odds from all the 20 bookmakers. However, this raised a problem as mentioned in Section 3.5, namely that this could be misleading. Further on we would need to get one account for each bookmaker in order to bet accordingly to our model, which is unrealistic. Hence, we decided to only use the four biggest bookmakers.

Validation

The span between a random-betting model and the zero line is several orders of magnitude which made it quite easy to validate both the Expected Goals- and the Elo model, since one of our goals was to analyze the possibility to make money. No matter how we tweaked the parameters and calibration period this models always ended up somewhere in between.

The Odds Bias model was much harder to validate. Everything from small changes in predictors to a change in the calibration periods had major implications for the result. Even though this model always, roughly, performed in the same order of magnitude, no matter our tweaks, 0.5 and 10 time the return during a year is a big difference. What made it so hard to validate was that, let us say we were to use 3 years for calibration that it only make sense to maximum use the two following years for validation which cannot be regarded as sufficient time span to draw any conclusions. This because the way bookmakers set the odds; the algorithms and the margin they use, changes over time which implies that the calibration period must be rather short, 2-3 years, and the validation period should be the following years.

Odds Bias

It is important to understand that the Odds Bias model does not predict the outcome of any games. It only checks if there exist biases in the odds and bets accordingly. This approach has its weaknesses, for example the bookmakers can raise the odds on a popular team in order to attract customers. For the Odds Bias model this implies that these odds will not be matched to the correct data points.

Another problem is that according to the Skellam distribution the probability of draw lowers when the expected number of goals increases even though the teams roughly got the same chance of winning. This might lead to that the Odds Bias models sometime over- or underestimates the probability of draw.

6.2 Improvements

Even though the Odds Bias model performs well, there is always room for improvements.

When it comes to validation, it would be preferable to develop a validation tool that could do a deeper analyze of the output. For example it would be useful if one could see the frequency of different bets and the corresponding gain as well as the variance.

We also think it would be possible to improve the preference of the Odds Bias model by merging it with the Elo- and the Expected Goal models in a suitable fashion. By merging it with the Expected goal model one might be able to make a more exact estimation of the draw frequency. The Elo-model could serve as an indicator to find outliers is the odds given by the bookmakers.