Explaining The Calculations Behind My Football Prediction Project

Zachary Tiller

Web App Link!

September 28, 2022

I. Introduction

For a while I have always been interested in how accurately football games can be predicted. Goals can come from seemingly nothing, and may be down to a rogue individual error which can greatly affect the outcome of a game. To me, this makes modelling football so interesting. When building out this project over the space of 1-2 weeks, amongst other projects, I always kept getting new ideas as means to refine the model, or make the software more efficient and flexible - I think the possibility for continual improvement is exciting and at the end I have noted down some future improvement ideas.

II. Model Overview

Overall, this model uses a Monte Carlo simulation to return probabilities of a Premier Leagues (Football) game outcome, broken down by scoreline (up to 6-6). To calculate the probability of a scoreline, I use a Bivariate Poisson distribution, where the input variables are the Home Teams Goal Rate, Away Teams Goal Rate, and the Covariance between them. I built an interactive web-app using streamlit - see the GitHub readme file for the link - and also a CLI for the user to interact in the terminal. The user has to specify the fixture (home away teams), how many games to look-back upon to whether to use G or xG to calculate the average goal rate parameter for each team.

I searched for APIs to gather football data from, but many features were paid and imposed limits on the number of API calls per day. However, I found one which I can use for gathering the data which I need. This dependence on an API is something which I would like to eradicate from my model in future because if the API stops being free then this will cause problems in the data gathering part of the programme - perhaps I can work on my own web scraping functions to manually collect data for me, but in the interest of gaining experience using an API and building an end-to-end model, I am happy with this for now.

III. Finer Details

As it is known that the goals scored in football games can be modelled by a Poisson distribution, I took this approach. Immediately though, I realised that the goals scored by each team in a game are correlated (to an extent); if a low-ranked team manages to score an early goal against a high-ranked team, they may sit back and defend deep to protect the lead, which may lower the amount of goals scored by the opposing team. So, instead of treating the home team and away team goal rate, say HGR and AGR, as independent random variables, I wanted to use a Bivariate Poisson Distribution to account for some dependency between them. There are now 3 input variables to this distribution:

- HGR, the home team goal rate (expected number of home team goals)
- AGR, the away team goal rate (expected number of away team goals)
- C, the covariance between HGR and AGR

This brought another challenge with it as now we have a 3rd parameter, the covariance. I calculate this as the actual covariance between HGR and AGR over the last L games (where the user can define the number of games to look-back). I cap the covariance, arbitrarily, at 0.3, because larger values lead to score-lines such as 6-4 being the most probable (which isn't quite correct!). However, this is probably not the optimal way to calculate the covariance.

Where this model allows for the most freedom is in the setting of HGR and AGR, the average goal rate. Usually, it is common for a baseline goal rate to be established, then small advantages (such as the home advantage, or attacking strength) and disadvantages (such as defensive strength of opponent) to be added and subtracted. I opted

for a different approach which I thought of myself, as I wanted to try and capture a teams recent goal-scoring form in this parameter. In determining this parameter:

1. The user specifies the number of games to look-back, L (this lets the user consider form)
2. The user then specifies whether to base the goal parameter calculations off of xG in each of those L games, or real goals scored in those L games. This is because xG provides a more realistic description of a teams true goal-scoring ability
3. Then, for each of the last L games, the goals scored are summed and divided by L to get an average
4. But, in each of those L games, the goals scored in game x are actually weighted by a unique weighting factor which aims to scale the goals scored by the difficulty of the opponent of that game. The formula for calculating Average Goal Rate, L games look-back is:

$$H/AGR = \frac{WF_x GF_x + WF_{x-1} GF_{x-1} + ... + WF_{x-L} GF_{x-L}}{L} \tag{1}$$

IV. Weighting Factor Explanation

To explain WF, weighting factor:

WF = (cumulative goals conceded up to game $x-1$ by opponent to be faced in the fixture to we wish to predict) / (cumulative goals conceded (cGA) up to game x-1 by opponent in game x) where *total games played* - $L <= x <$ *total games played* For example, imagine Arsenal are due to play Tottenham next, and we wish to look-back 3 games to predict the goal rates. Imagine Tottenham were Arsenals 7th game of the season, and, Arsenals opponents in games 4, 5, and 6 were Chelsea, Brentford and Bournemouth respectively. The weighting factors for the last 3 games would be:

- $\frac{cGA_{Tottenham\ game3}}{cGA_{Chelsea\ game3}}$
- $\frac{cGA_{Tottenham\ game4}}{cGA_{Brentford\ game4}}$
- $\frac{cGA_{Tottenham\ game5}}{cGA_{Bournemouth\ game5}}$

Now, the reason why we look at game x-1 for cGA is because, again, imagine when Arsenal played Brentford. This would be game 5 of the season, and so, because we multiply the weighting factor by the number of goals Arsenal scored against Brentford in game 5, we must use the cumulative goals conceded (cGA) by Brentford up to but not including game 5 because that reflects the defences strength just before they played Arsenal.

So, for further insight, for game 5 Arsenal played Brentford. Now, as Arsenals next opponent is Tottenham. Clearly the defences of Brentford and Tottenham are skilled differently; it is safe to say Tottenham has a better defence than Brentford (despite Brentfords good form at time of writing!). So, when calculating Arsenals average goal parameter, for when they play Tottenham, why should we use the goals scored Vs Brentford as a proxy for the expected goals scored Vs Tottenham, when Tottenham has a better defence so Arsenal would be less likely to score more goals? This is what the weighting factor tries to address; the (attacking) effort required to score 3 goals against Brentford would not be the same to score 3 against Tottenham

Note - this is a poor example as often London derbies can be high scoring! But, ignoring this fact, the weighting factor attempts to scale for difficulty of scoring against that team. In this case, up to but excluding game 5, imagine Brentford had conceded, cumulatively, 9 goals all season, whereas Tottenham only 4. The weighting factor, to multiply the goals Arsenal scored against Brenftord in game 5, would be $4/9$. So, if Arsenal scored 3 Vs Brentford, it is thought that Arsenal would have scored 3 x $4/9$ = 1.33 goals, if they had played Tottenham at that time. So, in this way, we calculate the **average goal rate for Arsenal Vs Tottenham** by considering the **difficulty scaled average of goals scored in Arsenals last L games** before the Tottenham match.

V. Conclusions

i. Shortfalls of the model

- We assume that the goal rate remains constant throughout the game - this implies that the probability of scoring in any minute is constant; this may not necessarily be true in practise. For example, a team chasing a 2-1 loosing position may be more likely to concede, thus increasing the goal rate of the attacking team
- The frequency which shorelines are observed in the simulation are very dependent on the Covariance (C) parameter; higher values lead to more observations of higher scorelines than lower - which is not the case in the real world. Football is generally a low-scoring games

ii. Ideas for future improvements

- Use market odds to get game probabilities

- Use live market odds to calculate the expected number of goals scored by each team, to then feed into my MC model to predict score distributions?
- Plot the distribution of each teams goals scored per game (the goal rate) infer the value of lambda (goal rate) which gives that distribution to the highest probability
- In terms of Bayesian Inference, you could set a prior on lambda to be the gamma distribution as this is the conjugate prior to a Poisson - then generate a posterior distribution over lambda by multiplying it by the likelihood (Poisson distribution, for the goals scored, x, in game n for all the lambdas). Or, take advantage of the conjugate prior and simply re-calculate the gamma distribution with slightly different parameters
- Add a time-decay factor to the calculation of avg goal rate; weight more recent observations heavier