Computer Science Tripos – Part II – Project Proposal

Predicting the outcomes of English Premier League matches

Joseph Marchant (jm2129)

Robinson College

12 October 2018

Project Originator: Joseph Marchant

Project Supervisor: Helena Andres-Terre

Director of Studies: Prof Alan Mycroft

Project Overseers: Prof A. M. Pitts & Dr R. Mantuik

Introduction

The largest betting market in the world is football, and the most popular league in the world is the English Premier League (EPL). I will be investigating 25+ years of results, and trying to predict their outcomes at an accuracy that rivals experts, and other literature on this subject.

A lot of literature uses post-match data, but those that do not tend to investigate potential features in little depth. I will be using only pre-match data, and will be investigating as many potential features as I can. The aim is to end up with a small subset of these features that can help my system predict well.

Data will be generated through a mix of web scraping (allowing me to compare and validate data from multiple sources), and manual collection where more appropriate. Each season has around 400 games, therefore my data will contain around 11000+ rows. In terms of features I expect to collect no more than 30 features worth of data, and expect my final data set to contain no more than 20.

I will be experimenting with multiple common classification techniques such as Logistic Regression, Support Vector Machines (SVMs), K-Nearest Neighbour (KNN), and exploring the use of Deep Learning. A strong focus will be on selecting the most important features, which includes testing out every feature I can collect data on. Through looking at the literature and pairing it with my own domain knowledge, I know this is something that can be improved on. I will use multiple evaluation techniques such as comparing accuracy and error rate to literature and experts.

Starting point

Literature can predict the winner of a match at a very high accuracy[1] given live match data such as the number of corners in a match at a certain time. *However* I wish to only use pre-match data, not live match data, as this would produce more interesting and useful results.

There are various data sets for a mix of pre-match and post-match data at football-data[2]. I may use some data from this, however all other data I collect will be via web scraping from multiple websites I decide upon.

I will be using Python to web scrape, clean data, engineer features, apply machine learning techniques, build Neural Networks and evaluate the systems. I have experience with cleaning data and minimal experience applying some basic ML techniques in Python. Everything else will require learning through books, tutorials and documentation.

I have some experience writing in LaTeX, which is how I will write my dissertation.

I have been a keen follower of the EPL my entire life, and therefore have knowledge of the domain and what features should help predict the outcome.

Resources required

For this project I will be using my personal laptop: ASUS ROG GL552VW. This has a quad-core i7, with a 256GB SSD and a 1T Hard Drive. The laptop is dual boot Windows 10 and Ubuntu 16.0.4. For larger programs I will be using Ubuntu, with Python 3 installed and using the VSCode IDE, but most of my code will likely be produced on Windows using Jupyter Notebooks with the latest Anaconda distribution. Most packages I will use are in the Anaconda distribution, however I may use further external Python 3 packages.

I will be using git for version control, and will have a public Github repository for code. Data and code will be backed up weekly to my external 1T Seagate Hard Drive. Should my laptop become unusable for any reason, I have an older version of the same laptop spare and should be able to get back to working on the project within a week if the situation arises. I require no other special resources.

Further Project Substance

This project will be treated as a three-way classification problem. The models will predict one of the following three options per row: Home Win (H), Away Win (A), Draw (D). Each row will represent one English Premier League match. The data set will cover the seasons from 1992/93 to 2017/18 inclusive. Each season has 20 teams, each of which play every other team twice: one at home and one away. This equates to 380 matches per season, giving the data set almost 10000 rows.

[1] http://publications.lib.chalmers.se/records/fulltext/250411/250411.pdf
[2] http://www.football-data.co.uk/englandm.php

Some potential individual features and sets of features include: Date, Home Team, Away Team, Result (target variable), Referee, Manager, Pitch Size, Form data (such as last 3 game win%), League Stats in current season (position in league of each team, win%, upset%, goals per game etc.), Injury Data, Last Season League Position, Transfer Window Spending, Player Data (such as total PL games experience of starting eleven, average win% of players, etc) and more yet to be decided upon.

Of course each bit of information used as a feature would need to be represented in numerical form in order to be useful to the model. I will investigate multiple ways of representing certain features. For example, say we are trying to represent the Referee column as numerical data. It could be represented as 'Referee upset%', i.e. % of games where this referee is present, and the underdog team wins, or it could be represented as 'Referee Home team win%', i.e % of games where this referee is present, and the specific current home team wins. These alongside other representations can be investigated.

I will start by collecting a minimal data set containing around 5 features but still using all 26 years of data, and then I will use some of the ML techniques above to evaluate how well they perform on it.

I first plan to try out Logistic Regression with this data set, as it is one of the simpler models to use. This should allow me to get a good grasp on building and evaluating the model, as well as giving me insight into effective features and test-train splits. A test-train split refers to the ratio used to split the data set into 2 smaller data sets; one used for training the model, and the other used for testing it.

I hope to get this first data set ready for testing before mid-November, as outlined in my Timetable. This should allow me to have a model built and evaluated by the end of Michaelmas term. My initial test-train split will be allocating around 6 years of data to testing (around 20% of the data), based on typical splits used in literature. However, this is a flexible ratio depending on further research and evaluation. Since my data is chronological I cannot use cross-validation to improve the reliability of my test results, which is something I must keep in mind. I will leave the current season data (2018/19) for further evaluation/testing and validation nearer the end of the project.

After this, I will investigate adding additional data columns/features to my data, and re-evaluate the algorithms (or move on to new algorithms), eliminating less-useful columns along the way. This will be an iterative process, allowing me to keep improving my data and models.

Some features will be pre-computed in order to take into account the dynamic nature of the data set (as all data is in chronological order), and the prediction model itself will be static. Furthermore, example dynamic features (also known as derived features) include any sort of 'Form' or 'History' feature. It may be the average win% over the last 7 games for the home team, or the win% at a certain stadium over the whole data set **so far** for the home team. This will allow my model to use past results and data to improve prediction accuracy, whilst also allowing me to investigate a wider range of potential features. Thus, a large portion of the work required for this project will be feature engineering.

Work to be done

The project breaks down into the following sub-projects:

1. Research literature and discover some useful sources of data. Investigate techniques to be used.
2. Collect data from numerous sources and clean/merge data sets for analysis and for inputs to any system.
3. Analyse features using data analysis and simple ML models, to generate an improved data set.
4. Build a more advanced prediction system. This is a project where I wish to experiment with multiple techniques and systems, and therefore by design I am uncertain of exactly which ones I will be applying at any given time. This extra flexibility is a positive aspect. This allows me to learn and improve my predictor as the project goes on, without being tied down to an overly-specific plan.
5. Evaluate prediction system(s).
6. Iterate the 'feature engineer – build model – evaluate – improve' cycle.

Success criteria

This is a notoriously difficult problem for algorithms to solve, and humans have always performed better. Famous expert pundits, Mark Lawrenson and Paul Merson, have demonstrated a 52%[3] accuracy over previous years in this exact problem: predicting the outcome (H,A,D) of each match. Therefore achieving a prediction accuracy rivalling this would be an extremely successful project. In light of this, my success criterion is to exceed a 40% prediction accuracy in this three-way classification problem.

Possible extensions

One key extension would be to compare my prediction system against more metrics and evaluation techniques, that give a further insight into the effectiveness of the system. Some examples include:

- Evaluate the accuracy of the model by simulating the process of placing a bet on every prediction the final system makes for the current (2018/2019) English Premier League season, and determining if a net profit was made.
- Compare my predictor to a statistics-based predictor. By this I mean using a probabilistic predictor that makes predictions based on a specific probability distribution that was determined from the data. For example, we could generate some values such as '% chance of home team win', and do the same for away win and draw, then draw predictions from this distribution. This would create a very insightful

[3] http://eightyfivepoints.blogspot.com/2017/07/how-are-lawrenson-and-merson-beating.html

and detailed set of evaluations, as a particularly accurate statistical predictor would highlight an effective feature.

Some other extensions include:

- Could split the data into two or more groups based on the difference in teams playing, applying the prediction models to each separately. For example, I could split the data set into two sets: The first covering all games where the difference between the league position of the home team and the league position of the away team is less than 7, and the second containing the games where this difference is 7 or greater. This could potentially help my model predict more accurately, as it may learn different weights in each scenario. For example, in the set where the difference is large the model will learn to nearly always predict the team with the better league position, whereas in the other set it will most likely be more complex.
- Have a Deep Learning model built and evaluated. This is something not certainly in my plan, but will be something I do if it becomes clear it is the best option. Therefore it is a useful extension to note.
- Extend the system to predict cup games between Premier League games; these notoriously have a higher percentage chance of upset, and this would be a fascinating phenomenon to explore.
- Extend the prediction system to other leagues, such as the English Championship (the second division). This may cause extra complexity if my original data sources for the EPL don't have the same data for these other leagues.

Timetable

Planned starting date is 18/10/2011.

1. **Michaelmas weeks 2–3: Oct 18 - Oct 24** Setup git code repository, and USB data backup. Research literature on sports prediction systems (both specific to football and the problem I am tackling, but also looking at more general problems and sports). Also research Python packages to use/ how to use them.
2. **Michaelmas weeks 3–4: Oct 25 - Oct 31** Determine good sources of data and collect this data via web scraping or whatever technique is most appropriate.
3. **Michaelmas weeks 4–6: Nov 01 - Nov 14** Clean and merge data, validating data from multiple sources along the way. Design initial data set in correct form to input into a predictor. Design first predictor (logistic regression). Begin potential feature research.
4. **Michaelmas weeks 6–8: Nov 15 - Nov 28** Train, test and evaluate the first ML system. Run a few evaluation techniques. Generate and collect data for some more features to add to data set, and decide upon next model to use. Begin building this next model.
5. **Christmas Vacation Part I: Nov 29 - Dec 20:** Continue building improved models and adding features in an iterative manner.

6. **Christmas Vacation Part II: Dec 20 - Jan 16:** Continue collecting more features, and experimenting with new techniques. By this point either have experimented with 3+ techniques, or have begun on a decided-upon larger system. Experiment with Deep Learning if time permits and I have not already.
7. **Lent weeks 0–2: Jan 17 - Jan 30** Narrow down on best system so far, for an in-depth evaluation. Run evaluation and write up a progress report.
8. **Lent weeks 2–4: Jan 31 - Feb 13** Work on next prediction system and data set depending on results and research.
9. **Lent weeks 4–6: Feb 14 - Feb 27** Continue working on this system and data set. Improve based on further feature analysis, evaluation results and research.
10. **Lent weeks 6–8: Feb 28 - Mar 13** Finish current version of system, run on latest data set and complete a full evaluation report. Begin writing dissertation.
11. **Easter Vacation Part II: Mar 14 - Mar 31** Experiment with improving system and/or looking at potential extensions, but mainly focus on dissertation.
12. **Easter Vacation Part II: Apr 01 - Apr 24** Complete draft dissertation, pass it to supervisor and DofS and tie up work.
13. **Easter term 0–2: Apr 25 - May 8** Finish up dissertation changes and additions.
14. **Easter term 3: May 9 Onwards** Proof reading and then an early submission so as to concentrate on examination revision.