Soccer Match Result Predictor

Catherine Gai

May 8, 2024

Abstract

In this report we propose a mathematical model in predicting the full time outcome of a soccer match, specifically on the dataset of the Premium League. We train the model based on historical match data on a variety of structures, including linear regression, multilayer perceptron, adaptive boosting (Adaboost), etc., and analyze the prediction accuracy on the match result between any two teams with data from a new match or with historical data. Our mathematical model yields an accuracy greater than 50% in a three class prediction (home win, away win, or draw).

1 Introduction

The prediction of sport match results have been a fascinating but difficult problem, due to its unpredictably complex dynamics.

In this report, we will discuss a mathematical model to predict the soccer match outcome. Specifically, given the historical match data of the home and away team, we would like to predict the outcome of a match, either with or without a new match data.

2 Model

2.1 General Model

For each soccer match, there is a home team and an away team, where each has a list of *statistics* associated with each team, as included in Tab.1.

The model is represented as a function f that takes a pair of team statistics of a match, x_h for home team statistics, x_a for away team statistics, and output the prediction outcome of a match. As the soccer matches we are interested all come from a league match, there are possible 3 outcomes: home team win, draw, and away team win, denoted as 1, 0, -1 respectively.

For a past match with home and away statistic x_h, x_a, and outcome y, it is obvious that we would prefer $y = f(x_h, x_a)$. Therefore, the model f is trained on the historical dataset of N matches: $\{(x_{h,i}, x_{a,i}, y_i) | i \in \mathbb{N}_+(N)\}$. However, we need to choose a a class of models for training purpose. Below we will introduce three classes: logistic regression (LR), multilayer-perceptron (MLP), or adaptive boosting (Adaboost)

Abbreviation	Description
xS	Number of shots
xST	Number of shots on target
xHW	Number of hit woodwork
xC	Number of corners
xF	Number of fouls committed
xFKC	Number of free kicks conceded
xO	Number of offsides
xY	Number of yellow cards
xR	Number of red cards

Table 1: Match statistics used in the model. All of them are integral ordinal features with non-negative values. x is H for home team (e.g. HS for home team number of shots) and A for away team.

2.2 Logistic Regression

A logistic regression is a multiclass classification scheme. Specifically, in case of n features and a set of classes C, a one-versus-rest logistic regression is given by a such of parameters $\{w_i \in \mathbb{R}^n | i \in \mathbb{N}_+(|C|)\}$ and $\{b_i \in \mathbb{R} | i \in \mathbb{N}_+(|C|)\}$ such that

$$f_i = \left(x \mapsto \frac{1}{1 + \mathrm{e}^{-(w_i^\top x + b_i)}} \right) \tag{1}$$

and the overall prediction is given by

$$f = \left(x \mapsto \operatorname*{argmax}_{i \in C}(f_i(x)) \right) \tag{2}$$

To find the parameters w_i and b_i, we prefer f_i to be a binary classifier on class i, which will lead f give the correct result. Specifically, for a data-point with feature x and label y, we would prefer:

$$f_i(x) = \begin{cases} 1 & y = i \\ 0 & y \neq i \end{cases} \tag{3}$$

Therefore, we can train $|C|$ individual binary classifiers based on the binary logistic regression. To train them, we use the logistic loss function, which takes in actual probability $p \in \{0, 1\}$ and predicted probability $\hat{p} \in [0, 1]$ of the label being class i:

$$L = ((p, \hat{p}) \mapsto -\hat{p}\ln(p) - (1 - \hat{p})\ln(1 - p)) \tag{4}$$

Therefore, the mean loss of the given dataset for each classifier is optimized in a limited-memory Broyden–Fletcher–Goldfarb–Shanno optimizer (L-BFGS) to compute the sequence of w_i and b_i.

2.3 Multilayer perceptron

A multilayer perceptron used in classification is, in essence, a multilayer logistic regression. Instead of the classifier being of form: $x \mapsto s_0(w^\top x + b)$ where s_0 is the element-wise sigmoid function, we suppose the classifier f takes the following form: $\forall x_0 \in \mathbb{R}^{n_0}$

$$f(x) = s(W_N x_N + b_N) \tag{5}$$

where $\forall i \in \mathbb{N}(N)$,

$$x_{i+1} = \sigma(W_i x_i + b_i) \tag{6}$$

where $(W_i \in \mathbb{R}^{n_i \times n_{i+1}} | i \in \mathbb{N}(N+1))$ are a sequence of weights, $(b_i \in \mathbb{R}^{n_{i+1}} | i \in \mathbb{N}(N+1))$ are a sequence of bias. Therefore, there are $N+1$ linear operations, with a total of N hidden layers. The hidden layer $i \in \mathbb{N}_+(N)$ has a dimension of $n_i \in \mathbb{N}_+$, which is the hidden layer size. n_0 and $n_{N+1} = |C|$ are determined by the input features and output classes respectively. σ is the activation function, for which we use the rectified linear unit (ReLU).

To convert the result to a probability distribution of n_{N+1} output classes, which is 3 in our case, we use the multinomial model where s is the softmax function: $\forall i \in \mathbb{N}_+(n_{i+1})$, the i-th element of $s(x)$,

$$s(x)[i] = \frac{\mathrm{e}^{x[i]}}{\displaystyle\sum_{j=1}^{n_{i+1}} \mathrm{e}^{x[j]}} \tag{7}$$

It is easy to verify that $\sum_{i=1}^{n_{i+1}} s(x)[i] = 1$. Therefore, we interprete $f(x)[1]$, $f(x)[2]$, $f(x)[3]$ are probability of home team win, draw, and away team win, respectively. We choose the cross-entropy loss as the loss function, which takes the actual probability distribution $p \in \{0,1\}^{n_{N+1}}$, and the predicted probability distribution given by the classifier $\hat{p} \in [0,1]^{n_{N+1}}$

$$L = \left((p, \hat{p}) \mapsto - \sum_{i=1}^{n_{N+1}} p[i] \ln(\hat{p}[i]) \right) \tag{8}$$

which is the multi-class version of the logistic loss function.

Therefore, if the number of hidden layers and the hidden layer dimensions are chosen, the classifier is specified by the set of weights and bias for each layer. These values can be learned by optimizing the cross-entropy loss function using the Adams optimizer.

2.4 Adaptive Boosting

Adaptive Boosting, or Adaboost, is an ensemble method that utilizes multiple week learners to form a booster classifier. In our cases, we uses decision trees with randomly downsampled feature selection as a week learner. The decision tree is favored due to the integral nature of our data, where the decision boundary is free to choose between any value between the consecutive integers. We can also take advantage of the fact that the discrete output classes are *ordinal*, meaning a home winning match predicted to be away winning should not be penalized the same as a home winning match predicted to be a draw. This can be adopted by slight modification of the binary decision tree training process as below, where we neither increase or decrease the weights that is predicted draw.

```
input  : The dataset of $N$ datapoints where datapoint $i \in \mathbb{N}_+(N)$ has feature $x_i$ and label
         $y_i \in \{-1, 0, 1\}$.
input  : Number of iterations $T \in \mathbb{N}_+$
input  : Initial ensemble classifier $F_0 = 0$
output: The final ensemble classifier $F_T$
$\forall i \in \mathbb{N}_+(N)$, $w_i \leftarrow \frac{1}{N}$;
foreach $t \in \mathbb{N}_+(T)$ do
    Train a weak classifier (decision tree) $h_t$ given the weights for data point $i$ as $w_i$ for
     $i \in \mathbb{N}_+(N)$;
    $\epsilon_t = \sum_{i \in \{i \in \mathbb{N}_+(N) | h_t(x_i) = -y_i\}} w_i$;
    $\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)$;
    $F_t = F_{t-1} + \alpha_t h_t$;
    Update weights: $\forall i \in \mathbb{N}_+(N)$, $w_i \leftarrow w_i \mathrm{e}^{-\alpha_t y_i h_t(x_i)}$;
    Renomalize weights: $\forall i \in \mathbb{N}_+(N)$, $w_i \leftarrow \frac{w_i}{\sum_{j=1}^{N} w_j}$
end
```

3 Analysis & Discussion

We present our analysis based on two prediction methods, using present match data and historical aggregated data.

3.1 Present Match Data

When using the present match data, we want to predict the outcome of the a match of two given teams based a current match statistics as outlined in Tab.1.

We trained all three models based on all historical match data from 2013 Season to 2018 Season, and validated them using 2019 Season to 2021 Season [1]. The validation accuracy of different models are included as Tab.**??**. Specifically, we trained the multilayer perceptron (MLP) with hidden layers of dimension 32, 64, 128, 64, 32 with a learning rate of 0.003, momentum of 0.9, and the Adaboost uses a decision tree as weak learner with depth 1.

It is expected that when going down the rows where models grow more complex and start incorporating ensemble model, the prediction accuracy gets higher. While a random there has been a significant boost to the prediction accuracy given that the a random guess gives a probability of 33%.

The predictive power of MLP is higher than that of the the logistic regression, since the collective behavior in the first N operation could be reduced to identity. Thus the performance is better. Adaboost, on the other hand, is an ensemble method that is more robust to overfitting.

In addition, the integral feature domain makes the base decision tree weak classifier particular well-performing. A small variation in the weight or bias may lead to a change on the predict

Model	New match data	Historical data
Logistic regression	46%	43%
MLP	51%	44%
Adaboost	58%	51%
Baseline	33%	

Table 2: The model performance comparison between using statistics of a new match or historical statistics.

probability. However, for a small variation in the decision boundary would not change the next decision node if the variation does not cross the integral boundary.

3.2 Historical Aggregated Data

Most commonly, we would predict the match result before the start of the match. Therefore, instead of using the present match data, we explore the use of the historical aggregated data. Specifically, to predict the match outcome of a home and an away team, we feed the classifier $\bar{x}_h$ and $\bar{x}_a$, where $\bar{x}_h$ is the mean statistics of the home team's previous home matches and $\bar{x}_a$ is the mean statistics of the away team's previous away matches.

The prediction accuracy for both prediction schemes is included in Tab.2. It is not unexpected that the performance degrade. The primary reason is that the historical aggregated might not reflected the condition during the actual match, so that there is a distribution shift between the previously trained data and the new match. In addition, the training data are all based on integral value. However, the mean aggreagate data is a continuous value in the set of real numbers. The parameter space between the integral value is unseen during the training process, so the inferred value might be drastically different from the integral inputs. There are ways in overcoming this, including data perturbation, structure filtering, etc.

4 Conclusion

In conclusion, we proposed a model to predict the outcome of a Premier League soccer match based on both new match statistics and historical aggregated statistics, trained on a variety of backbone structures including logistic regression, multilayer perceptron, and adaptive boosting. Generally speaking, the ensemble method Adaboost provides the highest performance, achieve higher than 50% accuracy in a 3-class classification task both using historical and new match data.

These statistics are definitely not an exhaustive list of influential factors to the match outcome, which we may include individual player statistics, etc.. In addition, the historical data do not genuinely reflected the current condition if a new match is played. Therefore, time-dependent models trained on datasets with larger feature sizes might further increase the predictive power of the soccer match outcome.

Acknowledgment

I, Catherine Gai, completed this report alone without help of others. The references included provide me with the data. Additional coding resources are the Python package documentations including NumPy, Scikit-Learn.

References

[1] *Football Betting - Football Results - Free Bets*. May 2024. URL: `https://www.football-data.co.uk/englandm.php`.