Georgia Institute of Technology

ECE 6254

Statistical Machine Learning

Final Report

Predicting soccer results in the English Premier League

Yifeng Jiang Nan Li Xia Wu Yujia Xie

Summary

Soccer is the most popular sport in today's world. Yet one of the reasons that soccer is beloved by many, is that it is highly unpredictable. It is always one main activity for gambling, and even best human experts could only predict 70% of the match results (home win, draw, away win).

In this project, we used conventional classification algorithms and matrix factorization algorithms to predict the match results of the English Premier League from 2008 to 2016, where in each year 20 best teams in England match with each other twice (one home, one away) throughout fall to the next spring. We could reach a 55% accuracy in this 3-class classification problem.

The topic is more involved as it seems – firstly, it makes no sense to evaluate a team's performance without considering its components' strength, making our observed data pairwise and fully dependent. Secondly, the composition of the team, injury and fatigue status, and whether the team is recently at its best constantly evolve. Also, the low total score nature of soccer and the unique outcome of "draw" add to the uniqueness of our classification/prediction problem.

1 Introduction

1.1 Background

Soccer is played by 250 million players in over 200 countries and dependencies, making it the world's most popular sport.[1] The reason why soccer appeals to us is that it is of high opposability, enjoyment and unpredictability.

In this project, we analyzed and predicted the results of soccer matches between teams in English Premier League during seasons 2008-2016. Every year in the nation's top-tier league, around 20 the country's best teams matched with each other twice. (That is, we will have 20*19=380 matches every year – or more precisely, every league season.)

1.2 Dataset

The datasets were sourced from two different places. One is from Kaggle[2], which contained more than 25,000 matches of 11 top-tier leagues in European countries from seasons 2008 to 2016, including match details like goal types (goals for and goals against), possession, corner and cards etc. It also contained players' attributes sourced from EA Sports' FIFA video game series. Another is from Sofifa website[3], which covered each team's yearly rating in 0-100. Note that, in our project, we only selected those related to the English Premier League.

2 Approaches

2.1 Classification

We first formulated this as a 3-class (Home Win/Draw/Away Win) classification problem. For each match (either in training or testing set), we could only use the teams' information and their past performance as features. We crafted features based on the following two assumptions:

- A team performing well recently tends to continue performing well in the coming match; when evaluating "performance", the team's opponents' strengths must be taken into account
- Team X with historical advantage when competing team Y tends to continue this advantage in their next match with each other

Thus for each of the 2 teams in a coming match, we evaluated the following "recent" features: the team's Goals For, Goals Against, number of shots on/off and number of league points gained in its recent 10 matches. For each feature, the 10 numbers were summed after weighted by opponents' latest FIFA ratings and a 0.9 time-discount factor. So there are in total $2 * 5 = 10$ features.

Between the 2 teams in a coming match, we further evaluated some "historical" features: the goal difference and number of shots on/off difference in their recent 4 matches with each other. Along with their latest FIFA rating difference, we got 4 additional features.

The Kaggle Kernel [4] introduced us to PANDAS library and helped us with the engineering side of the project, especially on preprocessing. But it included betting odds and line-up players as features, which somewhat violated the "prediction" setting in our option. We also thought it unnecessarily involved PCA and probability calibration. It didn't dig into the XML files for detailed match stats like shots number, and it only used the provided Kaggle database.

Since we have two datasets, we first merged teams' latest overall ratings to each corresponding match. Then we traced back historical matches to obtain those features mentioned above. To give each feature equal weight, we standardized all the features.

We tried six common classifiers: Random Forest, SVC, GaussianNB, AdaBoost, K-Nearest Neighbors and Logistic Regression. We randomly split 70% as training set and 30% as test set. The classifiers' parameters were set via grid search and evaluated via 5-fold cross validation.

2.2 Non-negative Matrix Factorization (NMF)

We also experimented matrix factorization (MF) approaches to specially address the pairwise nature of match data, that was only indirectly explored by conventional classification approaches. Another advantage of MF is that we can obtain predictions for many future matches at a time, while our classification approach requires the most recent match results, thus only works in a rolling fashion.

Consider the match outcome matrix X with entry X_{ij} being the goals team i scores when facing team j. In an ideal case, the scores between teams, say t1,t2,t3, are simply determined by the outer product of two vectors $B = [3\ 2\ 1]^T$ and $C = [1\ 2\ 3]^T$, i.e. $X = BC^T$. The first vector is the "offense values" of the three teams, with higher value being better; and the second is the "inverse defense values", with lower value being better. Then the 3 match results are t1:t2=6:2, t2:t3=6:2, t1:t3=9:1, as we expect (omitting diagonal entries).

Thus the idea is to approximate last season's X with a low-rank $\hat{X} = BC^T$, and use the "noise-free" $\hat{X}$ as a prediction for the results of this season. (Home/away matches are handled separately in two X as two teams face each other twice in one season, and that home/away matches tend to be very different.) If the difference between $\hat{X}_{ij}$ and $\hat{X}_{ji}$ is less than 0.5, we predict the match as a draw.

When constructing X from historical data, we tried two different ways to handle **missing diagonal values**: 1. just fill 0's; 2. eliminate diagonal elements from the Frobenius loss by maskings in the alternating optimization.

2.3 Poisson Probabilistic Matrix Factorization (PPMF)

2.3.1 From MF to PPMF

Probabilistic Matrix Factorization (PMF) is developed to naturally handle missing data and multiple occurrences [5] [6]. It assumes Gaussian noise of X on top of underlying B and C:

$$p(X|B, C, \sigma^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} [N(X_{ij}|B_i^T C_j, \sigma^2)]^{I_{ij}}, \tag{1}$$

where $N(x|\mu, \sigma^2)$ represents Gaussian distribution with mean μ and variance σ^2, and I_{ij} is the indicator of entry (i, j)'s presence in X, i.e. whether we have match data between team i and j. The regularization of B, C are achieved by their priors:

$$p(B|\sigma_B^2) = \prod_{i=1}^{m} N(B_i|0, \sigma_B^2 I) \quad p(C|\sigma_C^2) = \prod_{i=1}^{n} N(C_i|0, \sigma_C^2 I). \tag{2}$$

However, the Gaussian model inherently assumes the sport has high scores, e.g. basketball. While in soccer, we believe a discrete model with Poisson distribution is more suitable.

2.3.2 PPMF

To fit in the Poisson setting, we treat each game as a fixed interval of time, and assume goals are independent of time. The average rates of goals in different matches become the parameters we would like to learn. By replacing the Gaussian distribution in the likelihood of PMF, the likelihood function for PPMF is given by:

$$p(X|B, C, \sigma^2) = \prod_{i=1}^{m}\prod_{j=1}^{n}[\pi(X_{ij}|B_i^T C_j, \sigma^2)]^{I_{ij}} = \prod_{i=1}^{m}\prod_{j=1}^{n}[\frac{B_i^T C_j^{X_{ij}}}{(X_{ij})!}\exp(-B_i^T C_j)]^{I_{ij}}, \tag{3}$$

with the same priors as Equation (2). After some derivation, the log posterior distribution is:

$$\begin{aligned}\log p(B, C|X, \sigma^2) &= \log p(B|\sigma_B^2)p(C|\sigma_C^2)p(X|B, C, \sigma^2) + c1 \\ &= -\frac{1}{2\sigma_B^2}\sum_{i=1}^{m} B_i^T B_i - \frac{1}{2\sigma_C^2}\sum_{i=1}^{n} C_i^T C_i + \sum_{i=1}^{m}\sum_{j=1}^{n}[X_{ij}\log B_i^T C_j - B_i^T C_j] + c2,\end{aligned} \tag{4}$$

where $c1$, $c2$ are constants that do not depend on B and C. Maximizing the log-posterior over B, C with hyperparameters (σ_B, σ_C) fixed is equivalent to maximizing a objective function with quadratic regularization terms:

$$L(B, C) = -\lambda_B\sum_{i=1}^{m} B_i^T B_i - \lambda_C\sum_{i=1}^{n} C_i^T C_i + \sum_{i=1}^{m}\sum_{j=1}^{n}[X_{ij}\log B_i^T C_j - B_i^T C_j], \tag{5}$$

where $\lambda_B = \frac{1}{2\sigma_B^2}$ and $\lambda_C = \frac{1}{2\sigma_C^2}$. We could choose values for σ_B and σ_C by using soft weight-sharing method [7], but for simplicity, point estimates obtained from data is used in our code. Minimizing the objective function gives us a local minimum, which is a maximum a posteriori (MAP) estimate. PyMC3 is used to find the MAP estimate with Powell optimization.

3 Results

3.1 Classification

The baseline accuracy is about 45% if we "predict" all matches won by home teams. And our best accuracy reached about 55% with Random Forest classifier. Performances of other classifiers are similar, as shown in Figure 1.

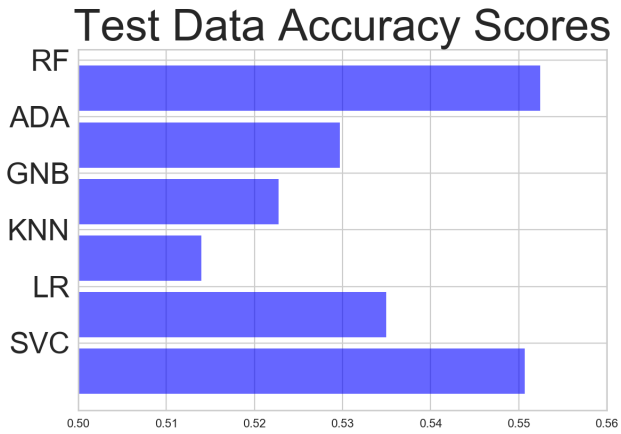

Figure 1: The test data scores of six different classifiers.

The confusion matrix of RF is shown in Figure 2. First we note that this classification problem is imbalanced since in actual matches, home teams won 46%, lost only 28%, the left being draws. While we correctly predicted about 87% of Home Wins and 54% of Home Defeats, this is actually due to "smart classifiers" always choosing to basically not predict draws, which is indeed really hard.

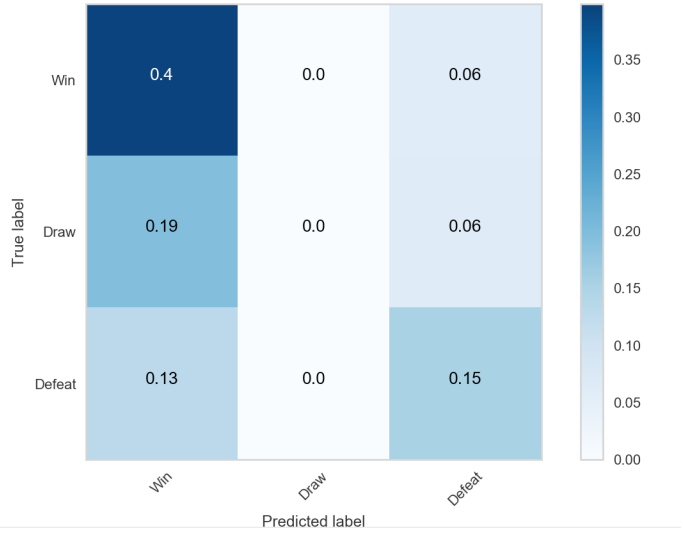

Figure 2: Confusion Matrix of the best classifier Random Forest.

3.2 NMF & PPMF

NMF with rank 7 reached a 46% accuracy on test set, barely exceeding the baseline. Figure 3 shows Poisson PMF works slightly better than Gaussian PMF, though both perform relatively poorly.

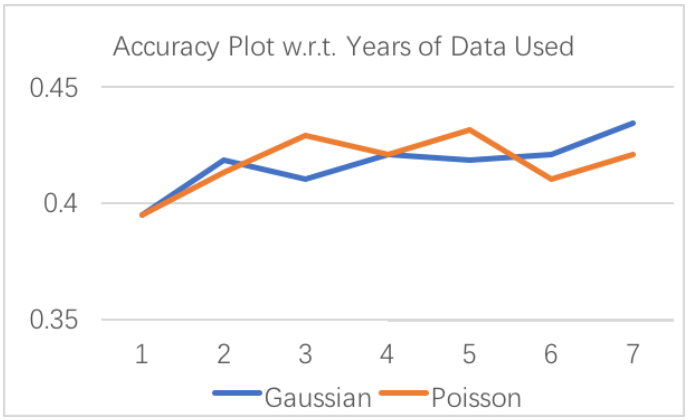

Figure 3: Accuracy plot w.r.t.years of data used

4 Conclusions and Future Work

After this project, we realized that predicting match results, especially capturing its pairwise and time-evolving nature, is hard.

We tried other techniques to improve accuracy, but gained little. We tried casting this problem into 5-class classification (Home win big, Home win small, Draw, Away win big, Away win small) instead of 3-class classification, and combining the results of Home/Away Win big and Win small for final prediction. We also tried oversampling the training set for balancing. Lastly, we tried involving more features like possession rates, and tried Bootstrap sampling for all classifiers.

For MF approaches, only the past game goals are currently used for prediction. More features could be incorporated as "side information" into MFs. [8].

Also, in the future we might adopt a way to calculate the probability more delicately: when the probability of winning or losing a game is close to 50%, we predict it as a draw. As the distributions of data are assumed, this should be no more than labor work.

5 Responsibilities

Nan Li, Xia Wu: Classifications
Yifeng Jiang: NMF
Yujia Xie: PMF

References

[1] F. O. Mueller, "Catastrophic head injuries in high school and collegiate sports," *Journal of athletic training*, vol. 36, no. 3, p. 312, 2001.

[2] "The ultimate soccer database for data analysis and machine learning." https://https://www.kaggle.com/hugomathien/soccer.

[3] "Sofifa: Top teams." https://sofifa.com/teams/topp.

[4] "Airback - match outcome prediction in football." https://www.kaggle.com/airback/match-outcome-prediction-in-football.

[5] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization.," in *Advances in neural information processing systems*, vol. 1, pp. 2–1, 2007.

[6] T. Tran, "Predicting nba games with matrix factorization," Master's thesis, Massachusetts Institute of Technology, 2016.

[7] S. J. Nowlan and G. E. Hinton, "Adaptive soft weight tying using gaussian mixtures," in *Advances in neural information processing systems*, pp. 993–1000, 1992.

[8] R. P. Adams, G. E. Dahl, and I. Murray, "Incorporating side information in probabilistic matrix factorization with gaussian processes," *arXiv preprint arXiv:1003.4944*, 2010.