Beating the Bookies: Predicting the Outcome of Soccer Games

Steffen Smolka – smolka@stanford.edu

Overview

Is it possible to predict the outcome of soccer games with high accuracy automatically? My project set out to investigate how far machine learning algorithms can be pushed, in a setting where outcomes are inherently random and human experts perform far from perfect.

Formally, I study the ternary classification problem of predicting the outcome $\mathbf{y}^{(i)} \in$ {home win, tie, home loss} (represented by +1, 0, and -1) of a game i from a feature vector $\mathbf{x}^{(i)}$.

Data Set & Features

As **raw data**, I used csv spreadsheets from [2] that contain the final outcome of 6500 Premier League games between 1993 and 2016. Although richer statistics exists, they are either limited to recent games or only available commercially at high charges.

From the raw data, I initially computed **goals scored** & **conceited** and numbers of **wins**, **ties**, and **losses** for the previous ω games for each team. These features proved ineffective. I then designed **fitness coefficients** to measure general team strength, offensive strength, and defensive strength. The coefficients are initially 1 for every team and get updated based on performance. Performing well against strong teams gives more points than performing well against weak teams. The coefficients satisfy two invariant: they are always non-negative, and their sum (over all teams) equals n (the # of teams) in each time step. Empirical evidence suggests that this invariant is crucial for their effectiveness.

I **did not use** team IDs or expert opinions in the form of betting odds as features (although I had the data available), because (i) I wanted to develop a universal model that works across leagues and (ii) I wanted to develop a model that can beat the bookies.

Models

Binary Classification. I began by studying the simpler binary classification problem

$$\mathbf{y}^{(i)} \,|\, \mathbf{y}^{(i)} \neq 0$$

i.e. the outcome of games that we know apriori to end in a win or loose.

SVM. I trained SVMs on both problems using libsvm. Ternary classification is handled by building a binary model for each pair of outcomes ("one against one").

I also trained my on SVN implementation using stochastic gradient descent, which surprisingly performed better. For the ternary problem, I classified games within distance $\gamma = 0.3$ of the hyperplane as ties.

Neural Network. I also trained neural networks on both problem using *scikit learn*. They support k-ary classification out of the box via the softmax function.

Discussion & Future Work

Discussion. Given the simplicity of the data set, the results are quite impressive. At the same time, there is lots of room for improvement. A surprise is that my custom SVM (even though it clearly suffers from overfitting) outperformed libsvm. In particular, it is surprising that my "hack", i.e. classifying games as a tie when they are close to the hyperplane, performs better than all other (presumably more principled) approaches in the ternary case. The biggest challenge of the project was (maybe surprisingly) extracting predictive features from the very simple dataset. All algorithms appeared sensitive to the addition of "bad" features.

Future Work. I would like to investigate algorithms that can **place bets** based on the models' predictions and some measure of confidence. My data set contains historical betting odds that can be used for evaluation.

	training err.	**10-fold cv err.**
home wins†	38%	38%
libsvm	33%	35%
custom SVM	10%	33%
neural network	15%	**27%**

Table 1: Binary classification

	training err	**10-fold cv**
home wins†	54%	54%
libsvm	51%	52%
custom SVM	13%	**47%**
neural network	42%	50%

Table 2: Ternary classification

†: the naïve model that always predicts *home win*

References

1. European soccer database. https://www.kaggle.com/hugomathien/soccer. Accessed: 2016-10-21.
2. Historical football results and betting odds data. http://www.football-data.co.uk/data.php. Accessed: 2016-10-22.
3. Chang, Chih-Chung and Lin, Chih-Jen. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
4. Karlis, Dimitris and Ntzoufras, Ioannis. Analysis of sports data by using bivariate poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52 (3):381–393, 2003.