Predicting Soccer Match Outcomes Using Machine Learning

Owen Chapman, Patrick Shao, and **Huu Le**
Pomona College

Abstract

A 1-sentence to 1-paragraph motivation for your work. (Why should anyone care about what you're doing?)

Predicting the outcomes of sports games is an absorbing interest for many people, from casual fans to analysts and statisticians to the sports franchises themselves. Accurate prediction of results involves a good model of sports teams relative strength, which can then be used to predict which team will win a given match. This model can be used by organizers to create fair tournament brackets, by analysts to determine rankings for a league, and by betting markets to determine the odds for a game. Applications such as these make sports modeling and prediction a multimillion-dollar industry.

Introduction

A 1-paragraph description of the problem you will solve. (To be turned into an Introduction section)

Prediction requires an algorithm to analyze historical data in order to generate a prediction of what is likely to happen next. In our problem of soccer match prediction, the goal is to predict the result of a match (win, loss or tie) between two known teams on a given date. The historical data will consist of various metrics estimating a team's match-day quality. These metrics may be based on the teams' past games, such as the winner of the last match between the two opponents. They may also include statistics about the match to be predicted that are public knowledge before the beginning of the game: for example, the players that are unavailable for the event or the odds given by betting vendors ahead of kickoff. Past successful algorithms have tuned a Bayesian network to generate predictions (Constantinou et al., 2013), implemented a Hidden Markov Model (Rue et al., 2000), and used a neural network (McCabe, 2008) to generate predictions.

Systems Description

A 1-3 paragraph description of how you plan to solve the problem. (To be turned into a System Description section)

To generate an accurate model of a soccer team, we first needed enough data for a machine to plausibly construct an accurate model of a team's relative strength. There are many online databases and APIs, both free and subscription service, that can be used to construct this model. The most complete database that we found for free was from football-data.co.uk, which gave us access to labelled match statistics for the last fifteen years of the English Premier League and about ten years of the Spanish La Liga. These match statistics include data about each game, such as goals scored, shots taken, and fouls for both teams, as well as the betting odds for several different online betting websites. These match data were compiled into training, validation, and testing sets for our classifier algorithms. We designated EPL seasons 2000-2009 as our training data set, EPL seasons 2010-2014 as our testing set, and LL seasons 2000-2004 as our validation set. Using statistics from La Liga for our validation set allowed us to increase the size of our training set to 10 seasons of EPL data, but we considered it best to train a classifier on the same league (EPL) as the testing set.

Before we could send our data sets to our classifiers, we had to generate historical feature vectors for each match, as opposed to the game-by-game statistics into which the data were originally organized. To generate these historical features, we sorted the matches chronologically and searched past games for relevant features. For example, one of our simple feature vectors consisted only of the result of the previous match between the two opponents. To implement this model, we iterated through every game in the data set, and for each game constructed a feature vector containing the result of the previous head-to-head match. This simple "last-result" approach became the baseline against which we compared more complicated feature vectors.

We compared several different feature vectors to the last-result baseline. These more complicated approaches can be described by combinations of three different modifications upon the baseline. The first modification was to increase the number of past matches considered: for example, from one head-to-head match to two or three. Secondly, we considered more historical information than head-to-head games, using a team's recent games against any opponent as a measure of "form," or how well the team is currently playing compared to its overall average performance. Thirdly, we altered which features we derived from each game. In our most simple case, we took only the result from historical games, but more complicated feature vectors took scores,

shots taken, shots on target taken, and other data from each game.

In addition to running different feature vectors through each classifier, we evaluated different classifiers themselves. The historical feature vectors we constructed were best suited to machine learning algorithms, and we evaluated a multinomial naive Bayes classifier, a support vector machine, a set of perceptrons, a stochastic gradient descent model, and a Gaussian naive Bayes classifier. These machines were implemented in Scikit, an open-source Python module for use in machine learning applications. We ran the algorithms using Scikit and built a shell program around Scikit to evaluate each algorithm's performance.

A model's success was determined by computing accuracy on the testing data set. The classifiers predicted each match as "win", "loss" or "tie", and accuracy was calculated as the fraction of games classified correctly divided by the total number of games classified. All classifiers were evaluated using many different feature vectors based on the three feature vector modifications described above.

Results

Algorithm	Base	Shots on Target	All Features
Gauss. Bayes	42.39%	46.28%	47.99%
Mult. Bayes	42.39%	47.37%	48.39%
SVM	42.39%	42.13%	44.96%
Perceptron	37.25%	25.66%	29.07%
SGD	33.02%	25.56%	25.87%

What results do you aim to obtain and how you will evaluate your approach.

When run on a testing data set, our models generated match predictions that can be compared to the actual match results. Our results for different algorithms were compared with each other to determine which approach classified the most labels correctly. We can also compare these results to external benchmarks, such as a random classifier or the predictions of human sports analysts. A low bar for success would be improvement upon the random classifier, and a highly successful implementation would be competitive with pundit predictions.

Conclusions

Summarize in a few sentences the most significant results of the results section After much data analysis, we noticed that often times comparing features individually seemed to provide quite low accuracies. Often times even combining multiple features such as the label of the previous game and goals of recent games would return accuracies that are slightly lower than comparing the features independently. That being said, when testing, the combined features appeared to be more resilient to changes made to the data set, as the percentages of the classifier remained relatively unchanged. We also noticed that the best indiciation of which team is going to win is dependent primarily on head to head games within the last 2 games, as any more than that produced more noise.

Return to the introduction. How well did we solve the problem? Relate our models to other models out there. Our primary goal was to create an algorithm that has the ability to predict future games. As compared to purely guessing, our classifiers perform significantly better. Both of our primary classifiers that we focused on (SVM with Kernel and Gaussian Naive Bayes Classifier) were able to produce accuracies that were higher than the baseline model.

If we had more time, what are some next steps? How do we think (conceptually) that we could generate more accurate predictions? (use betting odds in our predictions, use a DBN (time-consuming, would have to reconstruct our historical model), make some sort of less-noisy means of evaluating overall team worth than recent games against anyone. If we were given further time to develop, we intended to include betting odds within our calculations in order to get perhaps get more accurate predictions. It would be interesting to test this algorithm on real data sets with pseudo-money and observe whether or not a profit could be made with this machine.

If we were to develop this final project into a longer-term project, we could take several steps to improve our model. First, we would always welcome a larger or more accurate data set. Currently, our training set of ten seasons of the EPL contains almost four thousand games, but only two teams play in each game. As a result, the data set from which we construct a historical model of each team is actually no more than a few hundred games, and would be under a dozen if we only focused on a specific opposing team. Many teams in the Premier League also play matches outside the EPL, in tournaments or in lesser leagues, and this information would have been useful in expanding our training data set. Unfortunately, we were unable to find detailed match data for tournament games and did not have the time to integrate lesser league matches into our data set.

We could also improve our model by including in our feature vectors the published betting odds for the predicted match. These odds are generated either through popular consensus or by other predictive algorithms, and so would be highly correlated to the outcome of the match. Because of this correlation, we hypothesize that these betting odds would be weighted very highly by machine learning algorithms and be very influential in returning accurate predictions.

References

Constantinou, A. C.; Fenton, N. E.; and Neil, M. 2013. Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using bayesian networks. *Knowledge-Based Systems* 50:60–86.

McCabe, A., and Trevathan, J. 2008. Artificial intelligence in sports prediction. In *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, 1194–1197. IEEE.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-

learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Rue, H., and Salvesen, O. 2000. Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society: Series D (The Statistician)* 49(3):399–418.