Predicting Soccer Match Outcomes Using a Naive Bayesian Network

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 teams 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 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 statistics 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 ontaining 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, fouls, 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 Using this API, we are running several different algorithms to evaluate matches. We are starting with a simple Nave Bayes classifier, which uses calculated feature inputs from the previous season to predict match outcomes in the next season. We estimate the probability of the label as the number of results with that label in the previous season out of the total games played during that season. We likewise calculate features such as the probability of playing a given opposing team given the result using the previous years data. We plan to evaluate other models of increasing complexity using a validation set to test their relative accuracies. We hope to implement a series of perceptron classifiers to generate match predictions, as well as a hidden Markov model that reflects the results of recent matches when predicting the next match.

Results

What results you aim to obtain and how you will evaluate your approach. (To be turned into a Results section) When run on a testing data set, our models will generate match predictions that can be compared to the actual match results. Our results for different algorithms can be compared with each other to determine which approach classifies 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.

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.