UC BERKELEY

Soccer Match Prediction in the Serie A

Yannik Pitcan
April 30, 2016

1 ABSTRACT

I focus on applying statistical learning methodology to predict the results of soccer games in the Italian Serie A. Using past historical data from several years, features are engineered to capture the performance of a team. Support vector machines, logistic regression, and Adaboost are the best performing techniques with regard to predicting the results of future matches. The emphasis is on analyzing the tradeoffs between these methods' performances on our dataset with respect to precision/recall and error rates.

2 INTRODUCTION

Sports betting is a major industry in the United States and worldwide. With the advent of statistical learning methodology over the past few decades, inroads have been made into sports prediction in order to determine the outcomes of matches. However, most of this analysis has been done for baseball, basketball, and other games and sports where the game is less fluid. As an avid soccer fan, I decided to attempt to predict soccer match outcomes in the Serie A. The choice of the Italian Serie A was due to a few reasons. I believe the Serie A exhibits a parity among the teams that doesn't exist in other leagues. Also the tactical style of play in this league makes one think that past outcomes can influence the result of future games.

2.1 RELATED LITERATURE

Much of the work on this topic is done by bookmakers. Furthermore, they have access to private data which gives them a leg up on competitors. At Stanford however, a few students (Ulmer and Fernandez, Timmaraju) analyzed classifiers such as SVM, random forests, and

Hidden Markov Models for the English Premier League. My belief was that one could improve the feature selection methods used to get more accurate predictions on wins. In particular, ranks of teams were not always incorporated in the prior methods.

Another paper that was of relevance was work by Oyvind and Rue in their paper "Predicting and Retrospective Analysis of Soccer Matches in a League." Oyvind and Rue use a Markov Chain Monte Carlo (MCMC) approach to predicting the outcomes of matches. Although my work does not use a Bayesian dynamic linear model, my ideas for feature selection were influenced by both Oyvind and Rue and Ulmer and Fernandez. In particular, this influenced my decision to investigate a time dependent model of exponential weights when creating form and performance metrics for teams.

2.2 Dataset

The data used consisted of the results from the 2007 season to the current 2016 season. Seasons 2007 to 2014 were used for the training set and the 2015 and 2016 (ongoing) seasons were used for the test set. Our training dataset consisted of 3420 games while our test dataset had 320 games. For every game, the following data was recorded:

- Home team
- Away team
- Result (Win, Loss, or Draw)
- Goals scored by Home
- Goals scored by Away
- Shots taken by Home
- Shots taken by Away
- Corners taken by Home
- Corners taken by Away
- Home Win Odds
- Draw Odds
- Away Win Odds

Most of this data taken from http://www.football-data.co.uk/ and parsed via shell scripting. The betting odds were extracted from the Bet365 website.

3 Methodology

The main difference between my work and others is with regard to feature selection. I utilize odds ratios from betting agencies along with match statistics for prediction. Also, I apply time dependent weights in order to determine my features. The rest of this section covers more details about the specifics behind feature selection along with the challenges faced along the way.

3.0.1 Challenges

The majority of the difficulty came from the relatively inaccessible data for soccer as opposed to basketball and baseball which have troves of information. In order to attain data that includes player statistics per game, I would have to purchase this through Opta. Thus, I had limited data from which I could use for predictions. Most professional companies incorporate individual player statistics in their analyses, and also they have information about injuries as well.

Another obstacle I ran into was that a game like soccer has so many upsets that defy expectations as opposed to baseball or basketball. Many times a team with more skill loses to a weaker team due to luck. This is because there are so few scoring opportunities in a 90 minute match, especially at the upper echelons of the sport. For example, it is rare to have a repeat champion in the Serie A, with the exception of Juventus in the past few seasons.

3.0.2 Feature Selection

This was by far the most intensive task in this project because not only were new features necessary but I had to determine which features are relevant. In order to understand what features to engineer, one must ask the following:

- How can one capture the 'form' of a team?
- Is the team playing home or away?
- Has the team been scoring goals often? What is the overall performance outside of match results?

The concept of form is highly discussed in related literature. I experimented with two ways of calculating this.

One of my form calculations used time-dependent weights while the other one did not. As I did literature review, I noticed that time-dependency was not factored in when determining a 'form' feature. Thus I decided to give exponential weights to games played in the past.

For both calculations, one maps $\{win, draw, loss\} \rightarrow \{1, 0, -1\}$ in order to have a numeric value representing the result trend of a team as of late.

$$x_i = \begin{cases} 1 & , win \\ 0 & , draw \\ -1 & , loss \end{cases} .$$

In other words, assuming we use the past n games for match history, the two form features are calculated as follows:

If the game was played i matches earlier, then

- $$\tau(\{x_1,\ldots,x_n\}) = Form_{unweighted} = \frac{1}{n}\sum_i x_i$$
- $$\tau'(\{x_1,\ldots,x_n\}) = Form_{weighted} = \frac{1}{n}\sum_i x_i e^{-i}$$

Similarly, weighted and unweighted averages τ and τ' are applied to the statistics of corners, goals, and shots on target taken by the home and away teams to come up with our new features.

To encapsulate the home advantage, we subtract Away values from the corresponding Home values when calculating our features.

- Home team
- Away team
- τ(results in past n matches of home team) - τ(results in past n matches of away team)
- τ(shots taken in past n matches of home team) - τ(shots taken in past n matches of away team)
- τ(corners in past n matches of home team) - τ(corners in past n matches of away team)
- τ(goals scored in past n matches of home team) - τ(goals scored in past n matches of away team)
- Betting odds for home team winning
- Betting odds for draw
- Betting odds for away team winning

The determination of n was arbitrary. I used $n = 7$ because we thought that was the ideal number of matches representative of the form of a team. The idea behind this attribute is that we can explain using past results how confident the team is currently.

I believed the inclusion of betting odds would be an indicator of the ranks of the home and away teams so I used these as features as well. In the end, new feature tables were created for both the training and test data.

3.1 Models

Support vector machines, Adaboost, and logistic regression were used to train classifiers using 5-fold cross validation. The implementation of this was done in Matlab using the Classification Learner package.

3.1.1 Support Vector Machines

Past papers suggested that SVMs were the best method for game prediction. Thus I trained SVMs using the following kernels:

- Linear
- Quadratic
- Cubic
- Gaussian (radial-basis function).

Since I was dealing with a multi-classification problem (three categories were home team wins, away team wins, or a draw), I tried both the one-vs-all and one-vs-one approach.

3.1.2 Adaboost

The idea behind Adaboost is to aggregate weak learners through iterations on the training data to create a strong classifier. Intuitively, this sounded promising for game data where one can use decision stumps corresponding to features such as the average of goals scored in the past few games. I trained Adaboost classifiers using 30 decision trees of 20 maximum splits.

3.1.3 Logistic Regression

Along with SVMs and Adaboost, logistic regression was implemented as a baseline classifier.

4 Results

Performance of Classifiers			
Classifier	Accuracy	Precision	Recall
Linear SVM	AF	AFG	004
Gaussian SVM	AX	ALA	248
Adaboost	AL	ALB	008
Algeria	DZ	DZA	012
American Samoa	AS	ASM	016
Andorra	AD	AND	020
Angola	AO	AGO	024

We kept the classifiers that performed the best on the training datasets and ran them on the test feature set. Our best performing classifiers were Adaboost and RBF-SVM using feature data created by weighted match observations for the past 7 days. We create a feature dataset for the test set and running our SVM on this data, we were able to get about 44% accuracy predicting whether a result would be a win, loss, or draw. For Adaboost, we were predicting with 51% success. Considering that we would expect 33% success with randomly guessing, these results were fairly promising.

In the binary classification setting, RBF-SVM and logistic regression both performed very well with accuracy of 67% and 71% respectively. Tables recording the accuracies may be seen on the left hand side. The top table shows multiclassification accuracy whereas the bottom one shows results for binary classification.

Of particular interest was that the SVM setting did not predict any draws at all.

5 Conclusion