Data-driven insights into football match results

Kevin Bishop

Abstract:
The goal of this project is to use a variety of machine learning algorithms on a comprehensive dataset of football match results to predict team-agnostic match outcomes (namely, home win, draw, or away win). The dataset contains mildly sophisticated match data (e.g. beyond goals scored, but short of advanced statistics like chances created) for each side, as well as the match result, for the English Premier League from 2000-2012 (m = 4180). The test set is the equivalent match data from season 2013 (m = 208). The models presented include Multivariate Gaussian Naive Bayes (err = 59.13%) and Kernelized SVM using the RBF kernel (err = 52.40%).

I - Introduction:

Football is the world's most popular sport. It is played, watched, and enjoyed by billions of people worldwide. And in England particularly, the sport has more in common with religion than entertainment. With so much invested (both money and emotion) in the outcomes of professional matches in England, the same sorts of urban myths and armchair managers arise in English football fandom that we see in other arenas, like politics, the stock market, and the NFL. Having been a fan of Tottenham Hotspur for years, I am particularly excited to bring the power and insight of machine learning to bear on English football.

To be specific, the input to my algorithm is the difference between home team and away team previous-year averages in full-time goals, halftime goals, shots, shots on target, corners, fouls, yellow cards, and red cards (for and against). I then use two models, a Multivariate Gaussian Naive Bayesian model and a Kernelized SVM using an RBF kernel, to output a predicted match outcome, namely home win, draw, or home loss. Since goals are not perfectly deterministic, and a difference of one goal here or there can change a result dramatically, my expectations for accuracy were relatively low. But I was able to achieve meaningful accuracy on the three-class problem (win vs. draw vs. loss), even with relatively few features.

II - Related Literature:

Similar research in the area of soccer match prediction falls generally into two categories: time-series/Markov based, and non-time-series. The time-series models include that by Rue and Salveson, which uses a Monte Carlo time-series to simulate match results using only goals scored in prior games. Koopman and Lit use a bivariate Poisson distribution with coefficients varying over the course of the season according to a custom stochastic process.

The non-time-series research includes that by Goddard and Asimakopoulos, who studied the impact of match importance, distance travelled, and recent indicators of form on match outcome. But they focused on testing economic price efficiency of the betting markets rather than match prediction. Titman et. al. explore the interplay between yellow and red card issuance ("bookings") and goals in a

non-time-series manner, but primarily concern themselves with the effect of those bookings on the outcome of the game at the moment of occurrence, rather than predicting prior to match start.

My main source of background on this project comes from the CS 229 project by Ulmer and Fernandes (2014). They had less data available to them, both in terms of examples and features, and sought to do team-specific prediction rather than making team-agnostic predictions. For example, Ulmer and Fernandes sought to predict whether Tottenham Hotspur would win a given game they played, whereas I seek to predict any game correctly, regardless of participating teams. I also achieve a comparable error rate on the three-class win vs. loss vs. draw problem (52%) as Ulmer and Fernandes did for the two-class win vs. loss/draw problem (48%).

To arrive at the final data for use in the models, a fair amount of preprocessing was required. The raw features are ex post data about the match. Using such data renders the prediction problem uninteresting, as the data would only be available after the match, and a simple feature mapping of home goals minus away goals would produce a perfect classifier. So as to make the problem more interesting, I preprocessed the dataset to include the same 16 features listed above, but as average values over the previous season of play for each team and each statistic, respectively. This information would be publicly available prior to the match. Finally, I subtracted the away team averages from the home team averages to arrive at a 16-feature vector for each match. Below is a representative example from the data set, before and after preprocessing.

	RES	HT	AT	FT HG	FT AG	HT HG	HT AG	HS	AS	HST	AST	HC	AC	HF	AF	HY	AY	HR	AR
Raw	Draw	Spurs	Villa	0	0	0	0	8	11	2	2	6	5	12	17	3	2	0	0
Pro-cessed	Draw	Spurs	Villa	-1.32	0.39	-0.63	0.29	-2.21	4.29	-1.55	2.39	-1.32	1.42	-0.29	-1.61	-0.24	-0.47	-0.13	-0.11

III - Dataset and Features:

My dataset consists of the results and basic statistics from every match played in the English Premier League from season 2000-01 to season 2012-13. Training was done on the first 12 seasons (2000-01 to 2011-12, m=4180), and testing was done on the available data from the 2012-13 season (m=208). This dataset is publicly available from http://www.football-data.co.uk/data.php. Included raw features are halftime goals scored, full-time goals scored, shots taken, shots on target, corner kicks, yellow cards, and red cards for each team.

IV - Methods:

1. Multivariate Gaussian Naive Bayes

The first method used is a Multivariate Gaussian Naive Bayesian model. The parameters for this model are the probability of each outcome φ_y and the probability $\varphi_{x_j|y}$ of $x_j = x$ given a specified outcome y:

$$\varphi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=y\} + 1}{m+3}$$

$$\varphi_{x_j|y} = \frac{1}{\sqrt{2\pi\sigma_{j|y}^2}} exp\left(-\frac{(x_j - \mu_{j|y})^2}{2\sigma_{j|y}^2}\right)$$

for $y \in \{home\ win,\ draw,\ home\ loss\}$

This computation of $\varphi_{x_j|y}$ is from the gaussian PDF and is computed independently for each feature according to the Naive Bayes assumption of independence. The extra counts in φ_y of +1 in the numerator and +3 in the denominator are the Laplace Smoothing counts corresponding to a three-classifier. These parameters are used to maximize the Bayesian objective function:

$$y = argmax_y \prod_{j=1}^{n} p(x_j|y)\, p(y)$$

This amounts to choosing the label y that is most likely given the feature data for the given example.

2. Kernelized SVM with RBF Kernel

The second (and more effective, as we shall see) model is a kernelized support vector machine using the RBF kernel. At a high level, the SVM attempts to fit a hyperplane through the data that classifies each example correctly and with the greatest margin possible (i.e. correctly classified points are as far on their side of the hyperplane as possible). To do this, my SVM algorithm maximizes the objective function

$$J(\alpha) = \frac{1}{m} \sum_{i=1}^{m} L(K^{(i)\,T}\alpha,\, y^{(i)})$$

where α is a vector containing the weight on each training example, and $K^{(i)T}$ is the i-th row of the Gram matrix for the RBF kernel on the training data. The loss function L is the standard SVM loss function:

$$L(z,y) = max(0,\, 1 - yz)$$

After maximizing $J(\alpha)$ for the given step, I then update alpha according to a stochastic gradient descent rule

$$\alpha \leftarrow \alpha - \eta * 1\{y^{(i)}K^{(i)}\alpha \le 1\} * (y^{(i)}K^{(i)} - \lambda\alpha)$$

with learning rate η adjusted at each step, and regularizing term $\lambda = \frac{1}{16m}$. The algorithm was run 100 times over the dataset. Since the SVM is a binary classifier, I had to modify it a bit to work with the three-class problem. I trained the SVM on two binary classification problems: home win vs. draw/loss, and home win/draw vs. loss. The training and test datasets for each problem were identical. For final classification, I performed a logical and operation on the resulting vectors of predicted outcomes over the test set:

Win vs. Draw/Loss	Win/Draw vs. Loss	Final Classification
Win	Win/Draw	Win
Win	Loss	*
Draw/Loss	Win/Draw	Draw
Draw/Loss	Loss	Loss

The classification marked * is inconclusive, so my algorithm assigns any examples classified as such uniformly at random between the 3 classes. In practice, no test examples fell into this category.

V - Results:

Training set size: 4180 games

Test set size: 208 games

Table 1: Error rates

	Training Error	Test Error
Naïve Bayes	0.5787	0.5913
SVM	0.4787	0.5240

Table 2: Confusion Matrices

Naïve Bayes	Actual Loss	Actual Draw	Actual Win
Predicted Loss	14	11	13
Predicted Draw	13	18	24
Predicted Win	31	31	53

SVM	Actual Loss	Actual Draw	Actual Win
Predicted Loss	26	20	21
Predicted Draw	15	13	11
Predicted Win	17	27	58

Table 3: Precision

Precision	Win	Draw	Loss
Naïve Bayes	.4609	.3273	.3684
SVM	.5686	.3333	.3881

Table 4: Recall

Recall	Win	Draw	Loss
Naïve Bayes	.5889	.3000	.2414
SVM	.6444	.2167	.4483

VI - Discussion:

As expected, football match results turn out to be very hard to predict. But both of my algorithms made meaningful gains in accuracy over random guessing. Uniform random guessing produces an expected error rate of 66.67% on a three class problem, which both the Naive Bayes (7.54% more accurate) and the SVM (14.27% more accurate, see Table 1) easily surpass.

That being said, both the Naive Bayesian and SVM models have high error rates on the train and test data, as seen in Table 1 above. This is mostly due to the result of a match hinging on relatively rare nondeterministic events (goals). Furthermore, goals are not always indicative of how the overall match is going. For instance, one game pitted a home team averaging 1.92 more goals, 8.9 more shots, 4.9 more shots on target, and 3.9 more corners per game than the away team. The away team won the match anyway. This kind of result is what makes the sport so fun to watch, but it also makes life hard for prediction algorithms like mine. If just one effective counterattack over a 90 minute game, combined with some stellar defense and luck, can win the game in the face of statistical dominance by the other team, it's going to be very hard for either a probabilistic algorithm like Naive Bayes or even a non-probabilistic one like the SVM to predict an away win in such a scenario.

As it turns out, draws are the hardest outcome to correctly predict. Intuitively, this makes sense considering that wins and losses can be quite lopsided, while draws are by nature very tight affairs. For instance, games in which the home team has large positive differentials in goals scored, shots, and shots on target, or large negative differentials in fouls or red cards conceded will have very high probability given $y = win$, and very high margin in the SVM. In the Bayesian model, this is accentuated by the Bayesian prior probabilities, which are 47% for home win, 27% for draw, and 25% for home loss. Thus home wins are quite easy to predict. By contrast, a draw is always at a statistical disadvantage for prediction. Even if teams are similar statistically, games between such teams are as likely to go one way or the other as they are to result in a draw.

In the Bayesian model, the multivariate gaussian means for each feature given $y = draw$ are sandwiched between the corresponding means given $y = home\ loss$ on the low side and $y = home\ win$ on the high side. So in order to predict a draw, the features must split the difference between a win and a loss with little breathing room on either side. So it was somewhat surprising that the Bayesian model had comparable precision (33.3% to 32.7%, see Table 3) and markedly better recall (30.0% to 21.7%, see Table 4) for draws than the SVM model. This indicates that draws are better modeled probabilistically, as Naive Bayes does, than via separability, as the SVM does.

A shortcoming of the Bayesian model was its poor performance on home losses. One way I attempted to rectify this issue was to introduce a regularizing term in the SVM. I chose $\lambda = \frac{1}{16m}$ for my regularizer value based on experimentation on the training set. This yielded the training set error of 0.4787 (see Table 1). Larger choices of regularizer values tended to skew the predictions disproportionately towards home losses, while the opposite was true of smaller regularizer values. The optimal choice of regularizer value made the SVM much better at predicting home losses than the Bayesian model, nearly doubling recall for that class (21.4% vs. 44.8%, see Table 4).

Overall, the SVM performed better. It had better total accuracy and better precision on each class. The SVM also produced advantages in recall on home wins and home losses that offset its poor recall on draws. This was in line with evidence from my related literature, which suggested that the SVM would be the most effective method.

VII - Future Work:

Based on my findings, the main avenue for further work is in improvements to the input data. Currently, I use data from the previous completed season as a barometer of each team's strengths and weaknesses. But team composition varies widely between seasons. Players, managers, division placement, and even ownership can change from one season to the next. To improve the relevance of the input data to the match being predicted, there are two options. First, I could use the data from the season to date. This approach maximizes the relevance of the average, as it comes from the same group of players and the most recent timeframe. But the averages will fluctuate wildly for early-season games, and for the first game of the season, the models will be no better than a random guess.

Alternatively, I could use a rolling average of the 38 most recent games (a season is 38 games long in the Premier League). This will remove the fluctuation present in the first approach. But teams may change drastically in composition from season to season, meaning the average is taken over two distinct teams. The averages will also be difficult to compute. Finding the last 38 games for each team in the dataset would require indexing the dataset by team. And recomputing each team's average every game would be computationally intensive. Neither method is perfect, but either would be an improvement in relevance compared the current input data.

VII - References:

1. Goddard, J. and Asimakopoulos, I. (2004), “Forecasting football results and the efficiency of fixed-odds betting”. J. Forecast., 23: 51–66. doi:10.1002/for.877
2. Koopman, S. J. and Lit, R. (2015), “A dynamic bivariate Poisson model for analysing and forecasting match results in the English Premier League”. J. R. Stat. Soc. A, 178: 167–186. doi:10.1111/rssa.12042
3. 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: 399–418. doi:10.1111/1467-9884.00243
4. Titman, A. C., Costain, D. A., Ridall, P. G. and Gregory, K. (2015), “Joint modelling of goals and bookings in association football”. J. R. Stat. Soc. A, 178: 659–683. doi:10.1111/rssa.12075
5. Ulmer, B., Fernandez, M. (2014), “Predicting Soccer Results in the English Premier League”. Web. 21 November 2016. https://goo.gl/Zdj87n.
6. "Football Results, Statistics & Soccer Betting Odds Data". Football-data.co.uk. N.p., 2016. Web. 17 Dec. 2016.