Beating the NCAA Football Point Spread

Brian Liu
Mathematical & Computational Sciences
Stanford University

Patrick Lai
Computer Science Department
Stanford University

December 10, 2010

1 Introduction

Over recent years, professional sports betting has grown to be a multi-billion dollar industry. College football is one of the most popular sports that individuals bet on and has become a major contributor to the overall size of the sports betting market. There exists many different types of sports bets that can be made on college football games. The most popular one is the point spread bet. Point spread betting involves a number known as the "spread" which represents how much a team is favored to win by. For example, if a team has a spread of -7 then the team is favored to by win 7 points. Therefore, a person placing a bet on a team with a spread of -7 will only win if the team wins by more than 7 points. The purpose of the point spread is to create a fair betting market by offsetting *a priori* biases towards a given team. Point spreads are not static values in that sports bookmakers continually update point spreads to account for changes in the betting market. In theory, either team in a given match-up would have roughly a 50% chance of winning when the spread is taken into account. Expert sports bettors believe that achieving a consistent success rates of 60% or higher has become extremely difficult as bookmakers have begun to employ more sophisticated techniques in creating spreads [2]. Another type of sports bet is the straight bet. This is simply a bet on which team will win a given match-up without regard to the spread. In this project we focus on both the point spread bet and the straight bet.

The goal of this project is to learn, explore and apply modern machine learning techniques to a data set of NCAA Division I FBS football statistics in order to predict the outcome of a given match-up between two opposing teams. We wish to devise a binary classification system that can consistently predict match-up results at an error rate that is significantly better than random guessing and at a level that is competitive with expert sports bettors. Specifically, we will experiment with three popular binary classification algorithms: logistic regression, support vector machines (with different kernels) and AdaBoost (different variants). We test these algorithms by training models for predicting the winning teams in games with point spreads and in games without point spreads.

2 Data Set

We obtained our data set from two sources. The detailed team statistics for each game from the 2009 and 2010 seasons were scraped from `http://www.cfbstats.com`, a college football statistics repository. As the data is not provided in a format that can be easily downloaded, we resorted to downloading the entire website page-by-page for offline processing. We then created a HTML scraper in Java to extract, format, combine and transform the statistics into consistent and easily consumable format. The point spreads for each game from the 2009 and 2010 seasons were retrieved from `http://www.sbrlines.com`, a sportsbook aggregation site. We extracted the team identifiers, game date and the opening point spreads from this source and matched the point spreads to specific teams using the team identifiers and the game date, resulting in our raw data set. The raw data set is then processed with a Java program and used to generated training and testing examples. Finally, the examples are saved in a CSV format usable by our machine learning algorithms. Note that although we gather data for both the 2009 and 2010 seasons, we only use the 2010 season in experimenting with machine learning algorithms. The data from the 2009 season is used as initial values for computing examples for games that occur early in the 2010 season.

One problem we came across in gathering our data set was missing values in the data. However, we noticed that all of the missing values came from events which occur very rarely in college football games. These rare events usually don't affect the outcome of a game. We simply disregarded these rarely occurring events since we suspect that they would ultimately have very low prediction power in our models.

3 Features

The raw data set contains 94 fundamental college football statistics that captures each team's performance in a given game. Using the raw data set, we could easily compute new features and generate examples for feeding into our machine learning algorithms. Our initial examples were generated by first computing a cumulative average of the statistics in order from earliest games to most recent games. This gave us an average of a team's statistics up to a specific week in the season which could then be used in predicting future games the team plays in. This method of generating examples prevents us from making predictions on games using statistics generated in future games, which would distort our test error. For example, if we wanted to predict the outcome of a game at week k, we can use the teams' cumulative averages from week $k - 1$ (which we know doesn't incorporate future statistics) to train our model. For each game we create an example by concatenating the two teams' cumulative statistics up to (but not including) the date of the game. This resulted in a total of 730 examples with 210 features per example (features besides the fundamental statistics were added). Each example is labeled 1 if the home team wins and -1 if the away team wins.

Passes attempted	Interceptions	Points gained
Passes completed	Interception yardage	Points allowed
Passes intercepted	Passing touchdowns allowed	Red zone scoring percentage
Pass completion percentage	Passing yards allowed	Red zone field goal percentage
Pass rating	Rushing touchdowns allowed	Red zone touchdowns allowed
Passing touchdowns	Rushing yards allowed	Red zone field goals allowed
Passing yardage	Sacks	Third down conversion percentage
Rushes attempted	Sack yardage	Third down conversion percentage allowed
Rushing average	Tackles for loss	Quarterback hurries
Rushing touchdowns	Tackles for loss yardage	Passes broken up
Rushing yardage	Forced fumbles	Kicks or punts blocked

Figure 1: A subset of the fundamental statistics used as features

3.1 Momentum

One important aspect that can determine the outcome of a game is a team's momentum going into the game. Momentum shifts can occur abruptly and last for several games, e.g. a change in team strategy or an important player gets injured. We used two techniques to account for momentum in our models. First, instead of using a cumulative average of the fundamental statistics we use a k-game moving average of the fundamental statistics. Second, we computed 12 new features for each team that track momentum at different time horizons.

Average wins in last 3 games	Average losses in last 3 games
Average wins in last 5 games	Average losses in last 5 games
Average wins in last 7 games	Average losses in last 7 games
Average wins with the spread in last 3 games	Average losses with the spread in last 3 games
Average wins with the spread in last 5 games	Average losses with the spread in last 5 games
Average wins with the spread in last 7 games	Average losses with the spread in last 7 games

Figure 2: Momentum tracking features

4 Machine Learning Algorithms

In this project we experimented with three popular binary classification algorithms. Each of these machine learning algorithms are different in how they learn from data and will allow us to draw deeper insight into our problem.

4.1 Logistic Regression

Logistic regression is a very popular machine learning algorithm. It's simplicity serves as a good point of comparison for our other two machine learning algorithms. We used an implementation of logistic regression (generalized linear model) included in MATLAB's Statistics toolbox [7].

4.2 Support Vector Machine

Soft margin support vector machines are seen by many as the best off-the-shelf machine learning algorithm. We tested three of the primary SVM kernels: linear, polynomial (3rd degree) and radial basis function ($\sigma = 1$). In order to select the optimal SVM, we varied the box constraint parameter C and selected the value that gave us the lowest test error. This parameter selection technique also served to prevent overfitting the data. We used an implementation of SVM included in MATLAB's Bioinformatics toolbox [8].

4.3 AdaBoost

We experimented with two variants of AdaBoost: GentleBoost and ModestBoost. GentleBoost is a more robust and stable version of the original AdaBoost algorithm. Empirically, GentleBoost performs slightly better than the original AdaBoost on regular data, but is significantly better on noisy data and is more resistant to outliers. ModestBoost is a regularized version of the original AdaBoost that has better generalization. For our weak learner we chose decision trees (CART) with two split points. We boosted these weak learners for 800 boosting rounds (selected to prevent overfitting) to obtain our final classifier. Boosted decision trees select relevant features automatically, making it an interesting algorithm to use on our data which has a feature set that may contain irrelevant features. We used an implementation of GentleBoost and ModestBoost included in an AdaBoost MATLAB toolbox [1].

5 Evaluation Methodology

We estimate generalization error using hold-out cross validation where our data is split into 2/3 training set and 1/3 test set. All test set examples are from games that occur after games in the training set examples. This is necessary to ensure that we aren't distorting our generalization error estimate by training on data generated in future games. We also compare our results from the machine learning algorithms with random guessing, always predicting the home team will win and always predicting the away team will win.

6 Results

	Train Error	Test Error
Always Home	-	43.25%
Always Away	-	56.75%
Random	-	52.25%
Logistic	14.11%	38.52%
SVM (Linear)	7.77%	34.43%
SVM (Poly)	0%	35.66%
SVM (RBF)	0%	47.54%
GentleBoost	0%	27.46%
ModestBoost	9.41%	30.74%

(a) Results for straight bets

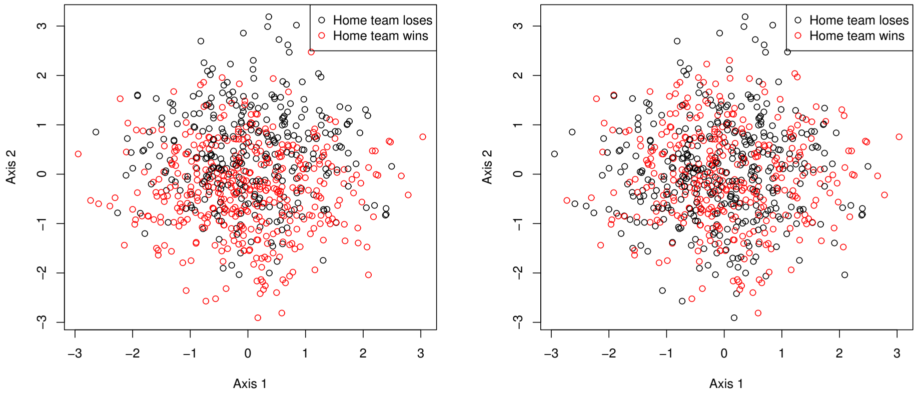
	Train Error	Test Error
Always Home	-	51.43%
Always Away	-	48.57%
Random	-	48.84%
Logistic	23.72%	50.41%
SVM (Linear)	23.52%	50%
SVM (Poly)	0%	44.26%
SVM (RBF)	0%	45.90%
GentleBoost	0%	48.36%
ModestBoost	0%	47.54%

(b) Results for point spread bets

Figure 3: Game outcome prediction results

7 Analysis of Results

Our results show that we are able to successfully predict the outcome of straight bets with an accuracy of 73% using GentleBoost. This result is slightly more accurate than previous work that has been done in predicting outcomes of other sports games [4][6]. In the case of predicting point spread bets, SVM with a 3rd degree polynomial kernel performed the best. We were able to beat random guessing by 4% and achieve an accuracy of 56%. This accuracy, which may seem low, is competitive with professional college football bettors who consistently achieve accuracies of 58% in betting on college football point spread games [3]. It is interesting that were able to achieve much higher accuracy in predicting the straight outcomes of games compared to games with points spreads. Intuitively this makes sense because the purpose of the point spread is to create a market such that either team has roughly a 50% chance of winning. However, we wanted to investigate our data further to determine why this may be the case. We plotted the outcomes of games against the first two principal components in order to draw insight into why we were not seeing better performance in predicting the outcomes of games with point spreads.



(a) Game outcomes without point spreads

(b) Game outcomes with point spreads

Figure 4: Game outcomes against top two principle components

Looking at Figure 4(a) we can see that there is a difference in the distributions of wins and losses from straight outcome games along the second principal component (y-axis). It seems that more losing labels are cluster towards the top and more winning labels are clustered toward the bottom. Thus, we would expect a binary classification algorithm to find a decision boundary and allow us to predict with good accuracy when we have a win or a loss. Figure 4(b) shows the top two principle components labeled with outcomes of games with point spreads. In this plot, it is almost impossible to differentiate between the two distributions as winning and losing labels are evenly mixed. Both groups of data seem to be similarly distributed along the top two principle component axes where variance should be highest. In this case any binary classification algorithm would have a very difficult time separating the data because, according to the features we defined, both groups are distributed similarly.

8 Conclusion

Though we were only able to beat random guessing by 4% in predicting point spread games, our results were still competitive with expert college football bettors. We believe the primary reason our results were not better is due to the set of features we used to train our machine learning algorithms. Alternatively, the fact that it is difficult to predict outcomes of games with point spreads may suggests that that market for point spread bets are "efficient" in that it is impossible to predict using historical data.

8.1 Future Work

An interesting avenue to explore in the future would be to work on constructing and discovering new features that can more accurately capture the effects of real game situations to the winning/losing margin of games. From our analysis of the first two principle components of the data, it is apparent that our current set of features does not capture enough information to allow us predict point spread bets no matter which binary classifier we choose. Some additional sources of data such as opponent rankings and play-by-play statistics may help in constructing features that will be useful for predicting outcomes of games with point spreads.

References

[1] Alexander Vezhnevets. *GML AdaBoost MATLAB Toolbox 0.3.* http://graphics.cs.msu.ru/ru/science/research/machinelearning/adaboosttoolbox

[2] Dr. Bob Sports. *About Dr. Bob.* http://www.drbobsports.com/about.cfm

[3] Dr. Bob Sports. *Best Bets.* http://drbobsports.com/analysis.cfm

[4] Hamadani Babak. *Predicting the outcome of NFL games using machine learning.* Project report for Machine Learning, Stanford University. http://www.stanford.edu/class/cs229/proj2006/BabakHamadani-PredictingNFLGames.pdf

[5] Kevin Gimpel. *Beating the NFL Football Point Spread.* Project report for Machine Learning, Carnegie Mellon University. http://www.cs.cmu.edu/~kgimpel/papers/machine-learning-project-2006.pdf

[6] Matthew Beckler, Hongfei Wang, Michael Papamichael. *NBA Oracle.* Project report for Machine Learning, Carnegie Mellon University. http://www.mbeckler.org/coursework/2008-2009/10701_report.pdf

[7] The MathWorks, Inc. *Statistics Toolbox 7.4.* http://www.mathworks.com/products/statistics

[8] The MathWorks, Inc. *Bioinformatics Toolbox 3.6.* http://www.mathworks.com/products/bioinfo