

Jonathan Pines
Nathan Marz
CS229 Final Project 12/16/05

Beat Cal: Machine Learning in Football Play-calling

Goals and Applications

Our overall goal is to successfully predict the offensive play-calling decisions of college football coaches. We will attempt to make such predictions on the basis of data from past games and from previous plays in the current game. While there is extensive literature on machine learning in some other sports such as baseball, football has been relatively neglected; furthermore, most machine learning applications to sports focus on game results and player performance rather than plays or coaching decisions. We hope that with this novel approach we will find new and useful results for the following two applications:

- (1) To assist a football coach in making defensive play-calling decisions appropriate to the expected offensive play.
- (2) To assist individuals in live sports betting, a service that casinos intend to introduce for football in the near future.

Data Acquisition

We got all of our game data from the Yahoo! Sports web site (<http://sports.yahoo.com/ncaaf/teams>). We wrote a (Java) program to navigate the site and find the URLs of play-by-play summaries for all games in the current NCAA Division 1-A football season.

We then wrote a (Ruby) program that takes as input the URL of a play-by-play summary, loads the page source, and parses the game data into a useful format for machine learning and in particular for the MATLAB 'load' function. The features we used for each play are as follows (each represented as an integer):

Features (input):

Quarter & time left
Score for each team
Down & yards to go
Position on the field

Classifications (output):

Play (run or pass)
Direction (left, right or center)
Yards gained

In the end, we focused our analysis on the Pacific-10 conference. For each Pac-10 team (Arizona, Arizona State, CAL, Oregon, Oregon State, Stanford, UCLA, USC, Washington, Washington State), we stored each game in a file with one line per offensive play for that team.

Data Analysis

There is myriad of questions we can hope to answer with this data. Given the input features for a specific game situation, for example, we might ask:

- (1) What play is most likely to be run next?
- (2) What plays are likely to be successful?

More generally, we might ask:

- (3) How much does a particular coach vary his strategy?
- (4) Which styles of play work well against which teams?

We would also hope to estimate our level of confidence in any answers we come up with, especially if the application is to betting.

In this project we will restrict our attention to the first of these questions – play calling – and leave the rest for future research.

There are also numerous machine learning algorithms suited to finding satisfactory answers to such questions. Because there is no literature on play-calling prediction, we had no solid basis for evaluating our results; our only practical benchmarks were human prediction and basic statistical data. Thus, we spent a good deal of our time exploring different possibilities using simple logistic regression to classify plays as runs or passes so as to explore how the features relate to one another and to the final play call.

In some situations, we found logistic regression to be surprisingly successful, and in fact, in many cases a simple linear model worked best. We also considered and implemented, with varying success, a few more complicated algorithms for a variety of outputs. In the end our best results came from a couple forms of logistic regression.

Logistic Regression for Binary Classification of Plays

Perhaps the most basic question to ask about the data is: can we predict, based on the input features, whether a given play will be a run or a pass? Our simplest attempt at answering this question was using logistic regression on what we expected to be the two most relevant input features: down and yards to go. We quickly realized that because dependence on downs is not linear, our results would be much improved if we trained Newton-Raphson – or any algorithm on any set of features, for that matter – separately for each down. With only one feature (yards to go) left, this basically amounts to picking, for each down, a cutoff in the number of yards to go between run and pass plays, i.e. the prediction is something like “pass with 3 or more to go, run otherwise.” With just this, in some situations for some teams, we were able to achieve over 80% accuracy, even when the statistics (i.e. go with the play that has been seen most often on that down) result in a much higher error rate. This is an indication that some coaches in some situations make decision in a very well-defined way, a fact that is not obvious from simply looking at statistical data on the ratio of run plays to pass plays.

One interesting piece of information we can extract from these initial data is the actual cutoff for different teams on different teams; that is, we can say, for example, that on 3rd down Arizona almost always runs with 1 yard to go and almost always passes with more to go. By comparing the cutoffs and errors, we can gauge both the consistency and the risk-taking propensity of different teams.

We also separately calculated the errors on running plays and passing plays. We found that the algorithm tends to misclassify a lot of passes on 1st down and a lot of runs on 3rd down. Combined with the fact that teams pass a lot more on later downs, this makes sense; the outputted line is heavily weighted by the type of play that occurs most often.

Next we added other relevant features to the binary classifier: time in game, score differential and position on field. One interesting trend we found was that while adding these features significantly reduces the error rate for 1st and 2nd down, it does not make much difference for 3rd down. This tells us that on 3rd down, coaches decide on the play primarily based on the number of yards to go for first down while on other downs they take more criteria into account. The error rates for logistic regression (on all features) with and without separate analysis for downs are as follows:

All downs trained together with down as a feature:

	ARIZ	AZST	CAL	ORE	ORST	STAN	USC	UCLA	WASH	WAST
1	0.419	0.419	0.334	0.422	0.429	0.391	0.409	0.393	0.344	0.382
2	0.305	0.359	0.355	0.457	0.371	0.284	0.4	0.414	0.349	0.498
3	0.195	0.176	0.256	0.258	0.268	0.409	0.313	0.218	0.311	0.263

Downs trained separately:

	ARIZ	AZST	CAL	ORE	ORST	STAN	USC	UCLA	WASH	WAST
1	0.394	0.427	0.331	0.411	0.402	0.385	0.371	0.417	0.325	0.379
2	0.291	0.347	0.329	0.434	0.393	0.288	0.317	0.379	0.344	0.44
3	0.188	0.122	0.215	0.2	0.251	0.417	0.254	0.218	0.227	0.322

Separate training of downs is better in a fair number of cases, although not as pronounced as we had expected. This is likely due to the increase in variance with a smaller sample size; training only on 3rd down means the sample size is less than 1/3 the size of training on all downs at once.

For comparison, the baselines (by predicting highest frequency play per down) are as follows:

	ARIZ	AZST	CAL	ORE	ORST	STAN	USC	UCLA	WASH	WAST
1	0.388	0.483	0.331	0.444	0.443	0.414	0.468	0.495	0.364	0.396
2	0.477	0.498	0.355	0.442	0.483	0.483	0.355	0.436	0.477	0.459
3	0.268	0.189	0.322	0.316	0.307	0.409	0.455	0.324	0.318	0.355

Logistic regression beats the baseline by a good margin in almost all cases. We also tried adding some new attributes that could be calculated from the features. Two examples are: average gain per run play or pass play in the current game, and play called on the last few plays. Surprisingly, most of these attempts gave us significant insight and improved error rates only slightly.

By comparing parameters found for different teams and testing error rates for a team's parameters on other teams, we were able to examine differences in play styles. With more analysis we could characterize these differences in concrete and useful ways.

Non-binary Classification

Next, we expanded the classification output to account for the length of the pass. The output was 0 for a run, 1 for a pass of under 10 yards, 2 for a pass of 10 to 20 yards, and 3 for a longer pass. Our approach to this classification problem was to separately apply binary classification to each output category as compared with all the other categories combined. That is, we separately found the probability that the play is a run as opposed to a pass, a short pass as opposed to a run or longer pass, and so on. The output of our algorithm was 4 separate weight parameters (theta) corresponding to the 4 categories. Given a new example, we compute the probability of it being each category using our four thetas that were computed. We first tried choosing the category that had the highest probability, but this turned out to be a bad approach. This is because of reasons similar to the voter's paradox; subdividing the pass category lowers the chance that any given pass category is correct, so that the algorithm predicts "run" almost exclusively.

We fixed this by doing hierarchical classification. First we used logistic regression as above to classify the data on just run vs. pass. Then, we took only the data for passes and tried to differentiate the three kinds of pass plays as follows:

For each category c of pass play

Set labels of pass plays with category c to "1"

Set labels of all other pass plays to "0"

Run logistic regression on the features with these labels and save the parameters found

Then, to make a prediction, we test the features against the parameters for each category and choose the category which outputs the highest probability. For pass plays, the results of this approach are:

	ARIZ	AZST	CAL	ORE	ORST	STAN	USC	UCLA	WASH	WAST
Regression:	0.4663	0.4837	0.428	0.384	0.455	0.437	0.475	0.452	0.494	0.4703
Frequency:	0.4663	0.577	0.503	0.459	0.484	0.484	0.504	0.466	0.533	0.4812

Frequency prediction just uses whichever pass distance has occurred most frequently in the past. For most teams, regression is a significant improvement to the frequency prediction; also, it is roughly evenly distributed across all downs and all categories predicted. We also tried predicting play direction (left, center, right) with little success.

Other algorithms

A few natural possibilities for algorithms and approaches to this task would be:

SVM with nonlinear kernel (we tried linear and got no good results)

HMM or CRF, perhaps with models for team attitudes as hidden states and situation/play pairs as observables. Thus in a given game the algorithm would attempt to find the most probable attitudes to explain a coach's current style of play, which would give a more accurate model for prediction. Such algorithms might do a better job of accounting for variations in a team's strategy over time and against different opponents.

Because we expect (intuitively and from examining the data) that dependence on most of our features is linear, especially for such features as yards to go, time left and

score differential, it seems that SVM and some other algorithms would be unlikely to provide better results than logistic regression. On the other hand, a completely different approach such as CRF might give completely new insights into the data.

Our expectation that dependencies are linear is consistent with our findings that in addition to having reasonably good error rates logistic regression makes intuitive predictions in many situations. As a simple example, it predicts run on 3rd and very short, and pass for very long plays. Getting down in score makes it predict pass a lot more, which also makes sense.

Analysis: Where does the error come from?

We have certainly had some success in prediction, but most of our predictions still have what might seem to be large error rates. There are a number of reasons why the error in these predictions is so high. First and foremost, there is inherent randomness in how a coach makes his play-calling decisions. No coach follows a well-defined formula for decision making, and furthermore, one of the primary intermediate objectives of a good coach is to gain an advantage by surprising the opponent. Nevertheless, we believe there is much room for improvement in play-call prediction, and particularly in the realm of machine learning, as discussed in the following section.

Future Possibilities and Gambling Application

We have only begun to scratch the surface of possibilities for play prediction and analysis in football. There seems to be virtually no literature on the topic, but perhaps unjustly so; we believe that with more detailed data about players, formations, and other factors, as well as different algorithms, models and approaches, very accurate play-predictors could be built.

Recently several major casinos in Nevada and elsewhere have announced their intention to initiate a program for live betting on college and professional football games. This would include exactly the kind of scenarios we are analyzing, namely, betting on individual plays. A casino gambler could thus use our analysis to place bets on plays that he knows to be likely where other gamblers using personal knowledge, intuition and statistics might get the odds wrong.

Summary

It is clear that a program with the ability to predict play-calling better than a knowledgeable person with access to statistics would be of great value to a coach. Any information a coach can get about the likely behavior of his opponent can provide a crucial edge, since he can call a defensive play that will cover the likeliest possibilities. The application to gambling is even more direct; if a person knows that the probability of a play occurring are even slightly better than the odds given by the casino, he stands to make tremendous gain.

As a final point of interest, our results show no well-defined correlation between predictability of play-calling and success of the team. We are able to predict great teams such as USC and poorer teams with comparable accuracy, and in the future, we hope that this fact can be used to tip the odds in favor of the former underdog and perhaps help Stanford to win Big Game.