data golf blogs
The DG archives

FEBRUARY 14, 2017 BY DATA GOLF

A Predictive Model of Tournament Outcomes on the PGA Tour

[Last Updated: Feb. 12, 2018]

(Updated version of this document can be found here)

The purpose of this model is to provide probabilistic forecasts for outcomes of a PGA Tour event. Specifically, we focus on four outcomes: 1) Winning, 2) Top 5 Finish, 3) Top 20 Finish, and 4) Making Cut. Therefore, for each player in the field, the model assigns a probability for each of events 1) – 4) being realized. The note below details the features of the model.

Brief Summary *(skip if you plan to read this article in its entirety)*

These are the bare essentials to understanding what matters in this model. The most important player characteristics in the model are different measures of adjusted scoring average (adjusted for course difficulty and field strength). We use a 2-year average, 2-month average, and the player's most recent event average. The model weights the averages calculated from longer time horizons much more heavily. The number of rounds played by a player also factors importantly into the model; more weight is placed on scoring averages calculated from a larger sample of rounds. Finally, the standard deviation of each player's sample of scores is important. Players with a lower standard deviation are more consistent in the model simulations.

Model Overview

Here we provide the general method through which the model generates the probabilistic forecasts, saving the details for later. In the model, each player's score can be decomposed into a "predicted component" and a "random component". The former term is determined based off of the observable characteristics of each player, while the latter term can be thought to capture all unobservable factors that affect a player's score on a given day.

Probabilistic forecasts are produced by simulating the outcome of a tournament many times. I'll first go through what a single simulation entails. For simplicity, suppose we are interested in predicting a 1 day tournament. The model first generates the predicted component to each player's score using observable characteristics (scoring average in the past 2 years, for example). Then, the simulated score for each player is this predicted component plus a random term that is drawn from a probability

distribution (for example, a normal distribution with mean 0 and some variance). With each player's simulated score in hand, we can determine the finish position of each player in this simulated 1-day tournament. By repeating this simulation process many times, we can determine, for example, how many of the simulations were won by each player. The win probability for each player will then be calculated as the number of simulations where that player was the winner divided by the total number of simulations.

So, what changes in each simulation? That is, why doesn't the same player win every time? In each simulation, the players' predicted component is always the same, but the random component is, in general, different every time (recall that it is a draw from a probability distribution). For example, while Jason Day will always have an excellent predicted component to his score, in some simulations he will receive a "bad" shock, while in others he will receive a "good" shock. The random component of a player's score is meant to reflect the random variation we observe in all golfers' scores from one round to the next.

While we focus on the four outcomes defined in the first paragraph, the probability of any outcome of interest could be computed from these simulations.

Data and Estimation

PGA Tour, Web.com Tour, and European Tour data at the round level from 2013-onwards is used to estimate the model. The variable that we are actually predicting is an adjusted score. This adjustment accounts for field strength as well as course difficulty. Following Broadie and Rendleman (2012), we adjust scores by first running the following fixed effects regression:

$$Score_{ij} = \mu_i + \delta_j + \epsilon_{ij}$$

where i is indexing player, and j is indexing a specific year-event-course-round. In practice, this is equivalent to regressing score on a set of player dummies and year-event-course-round dummies. From this regression, we collect the δ_j terms, and then define the adjusted score as $Score_{ij} - \delta_j$. The interpretation of the δ_j terms is the tournament-course-round scoring average after adjusting for the strength of the field (the μ_i terms). The separate estimation of these fixed effect terms is made possible by the fact that there is overlap between the sets of players that participate in each tournament. That is, if there was a tournament field made up of players who did not participate in any other tournaments, we could not estimate the δ_j terms for that tournament-course-round, or the μ_i terms for those players. Broadie and Rendleman (2012, p.7) provide a simple example to understand this intuition. (In practice, we actually estimate this fixed effects regression in two-year windows; essentially fixing each player's ability level in each window).

To get a sense of the magnitude of this adjustment, here are two of the biggest adjustments made for field strength in our sample. In the first round at the 2013 Tour Championship, the field scoring average was 69.1. However, our adjusted scoring average (the δ term) was 70.3. This difference (70.3 - 69.1 = 1.2 strokes) reflects the strength of the field. Conversely, in the first round of the 2016 Joburg Open, the field scoring average was 71.0, while our adjusted scoring average was 68.9. This difference of -2.1 strokes reflects the weakness of the field at the Joburg Open. These are two of the more extreme differences we obtain between the raw field average and the adjusted average; the majority

of the adjusted scoring averages are within 0.5 strokes of the raw field average (although, a good portion of the European Tour events have adjustments of more than -0.5 strokes).

From here on, we simply use "score" to refer to the adjusted score (defined as raw score minus the adjusted average for any given day). Recall that to simulate a player's score, we need what I have called the predicted component and the random component to their score. First, let's look at how the predicted component is estimated.

Predicted Component

The estimation technique used is OLS (ordinary least squares) linear regression. (The choice of this simple estimation technique is discussed in footnote 1). Therefore, the predicted component of a player's score is just (loosely speaking) the conditional expectation of their score given a set of observable characteristics. The main observable characteristics used are:

- 2-Year Adjusted Scoring Average
- 2-Month Adjusted Scoring Average
- Previous Event Adjusted Scoring Average
- Days Since Last Event

Using OLS, we estimate the effect of these variables on a player's expected score on a given day, and through this obtain the predicted component to a player's score. We have considered using course-player specific effects (certain players perform well on specific courses). However, the bottom line is that other than the listed variables above, it is hard to find other variables that have any predictive power on a player's score.

There are many complexities that arise in practice. Most importantly, there are many players for which we have only a few data points on their past performances (or none, in the case of rookies). Naturally, we do not take a player's past scoring average as seriously when it is the product of only a few rounds. To account for this, we include an interaction term between the number of rounds played in the past 2 years and 2-year scoring average. This allows us to estimate how much a player's scoring average should be discounted when it is made up of only a small number of rounds. For example, a player with a +3 strokes-gained average over 10 rounds will not receive as high of a prediction as a player with a +3 strokes-gained average over 100 rounds. In general, the fewer rounds you have played, the more your "effective scoring average" gets pushed towards zero.

Further, in practice, we end up estimating a set of regressions; one for rookies, one for players that have not reached a certain threshold of past rounds *and* have not played recently, one for players who have not reached a certain threshold of past rounds *but* have played recently, and then a main regression with all players who have played over a certain threshold of rounds. The rookies regression includes just a constant; therefore our predicted component for rookies is simply the average score of rookies in the estimating data.

The general idea behind having these separate regressions is to penalize players slightly who have not played enough rounds for us to be confident in their true playing ability. This penalty basically shows up in the constant term of each regression (so, in the main regression, the constant term should be

zero; we end up with an estimate very close to zero). Evidently, the choice of a threshold level for past rounds played is a critical one. At the moment, it is set at 100 rounds. This determines which regression we use to generate any given player's predicted component. It may seem like this would be an arbitrary decision. However, that is the nice thing about estimating a predictive model; we simply make our decisions based on what provides us with the "best" predictions, where we define what is meant by best later.

Random Component

At this point, we have the predicted component to a player's score. Recall that, in each simulation, we draw a player's random component from a probability distribution. But what should this distribution be? We decide to use a mean-zero normal distribution for all players, but allow players to have individual-specific variances. In our estimating data, we can recover the random component to a player's score by taking the difference between the player's actual score and their predicted score. In the regression context, this is the residual. Then, to estimate player-specific variances, we simply pool all the residuals for a given player and estimate the sample variance of these residuals. Allowing the variance to differ by player reflects the fact that certain players have less variability in their scores than others. This plays a meaningful role in the simulations. Again, there is the problem of having only a few observations for some players. And, again, the solution is to only estimate player-specific variances for those with a sample of past rounds above a certain threshold. For those below the threshold, we group the players and estimate a single variance for all players in the same group. For example, a single variance is estimated from all our data on rookies, and, as a consequence, the random component to a rookie's score is always pulled from the same distribution. Further, there is still the question of whether player-specific differences in variances reflect *true* differences, or just statistical noise (even with a big sample size, say, 100 rounds, there can still be a lot of noise present). To test this, we aggregate our data to the player-year level (only for those players who have played at least 50 rounds in a year) and estimate the relationship between a player's estimated standard deviation in scores from one year to the next. If a player is truly a high standard deviation player, then that should be the case in each year of our data. (Basically we are asking, "does having a high standard deviation in one year mean you will have a high standard deviation in the next year?"). We find that there is substantial regression to the mean (as expected..), and as a result the player-specific variances we use in practice are discounted towards the mean standard deviation of scores on the tour.

An additional complexity is the possibility that, within a tournament, players' random components will be correlated across days. That is, a good shock on Thursday is more likely to be followed by a good shock, than a bad shock, on Friday. We quantify the persistence in random shocks by estimating an AR(1) process using the aforementioned residuals from our regressions. An AR(1) process is one where, in a time series, the current realization is a function of only the previous realization (and not any realization before that). Thus, in this context, a player's random component in round 4 depends only on his random component in round 3 (and not that in round 1 or 2). In practice, we estimate this coefficient to be quite small (1-2% of a player's "residual" performance carries over from round-to-round).

We now have all that is necessary to simulate a full 4-day tournament. The predicted component of a player's score is the same for all 4 days (any information that is contained in a player's performance on day 1 of the tournament is captured by the persistence in their random components from day 1 to day

2, day 2 to day 3, etc). For each day of the tournament, each player's random component is drawn from their respective distributions and added to their predicted component. Through this we can obtain their 4-day total. At the halfway point of the tournament the cut can be made to the top 70 players and ties. Repeat this procedure 15,000 times and we have our probabilistic forecasts for any outcome of interest!

Model Evaluation

In any predictive exercise, it is critical to check that the model is generating reasonable predictions, and to determine whether adjustments to the model can be made to generate better predictions.

We evaluate the model in two ways. Firstly, we focus just on the predictive component. In the predictive setting we are in, we would like to minimize the difference between players' actual scores and their predicted scores generated by the model. A simple summary measure is the mean-squared prediction error, defined as:

$$\frac{1}{n}\sum_i (Score_i - Predicted_i)^2$$

where i is indexing the unit of observation (a specific year-tournament-round-player observation), and n is the sample size. The smaller is the mean-squared prediction error, the better our model is fitting the data. It is important to evaluate the model using data that was not used to estimate the model (for reasons such as overfitting). In practice, overfitting is likely not an issue in this setting simply because we cannot explain that much of the variation in daily scores with the model. We estimate the model only using data for 2005-2015, and evaluate the model using data for 2016. The choice of which regressors to include in the regressions, as well as the most appropriate choice of threshold level, is informed by their respective effects on the mean-squared prediction error.

The second method through which we evaluate the model is what has been termed "calibration" among certain forecasting pundits. Given that our model generates probabilistic forecasts (i.e. Player A has a 2.6 % chance of winning this event) it should be the case that the forecasts actually match up with reality. That is, when the model says an event should happen x percent of the time, we would hope that in reality this event does happen roughly x percent of the time. The main limitation to this method of evaluation is sample size. If the model is predicting certain events to happen only a small percentage of the time, then we need to have many predictions that fall into this range to get an accurate sense of how closely the model matches reality. Using the 2016 data, we have 5625 observations at our disposal (this is the number of tournaments * field size). We group predictions for our four outcomes of interest (winning, top 5, top 25, cut) into small categories defined by their probability of occurrence (ex: 0% < Prob(Win) < 3%, or 42% < Prob(Made Cut) < 45%). Then, for those players with predictions that fell into a certain category, we calculate what the actual probability turned out to be. For example, in 2016, the model predicted a win probability (for a player at a tournament) between 3% - 6% 149 times. The actual win probability for this group of players in 2016 was 4% - therefore the model did quite well in this instance. In general, the categories for which we have a large number of predictions (for example, many players have predicted win probabilities in the range of 0%-2%, or cut probabilities in the range of 50%-52%), the model forecasts match the data quite nicely.

This note will be continually updated as we make changes to the model - it is still a work in progress.

Footnotes

1. Given the massive advances in predictive algorithms in the machine learning literature in recent years, it seems like there could be much to gain by adopting one of these more complex techniques in estimating the predicted component in our model. However (keeping in mind that my opinion is certainly not an expert one in this area), I think there is little to gain by adding complexity in this specific setting simply because there is not a great deal of variation in daily scores (our outcome variable) that can be successfully predicted. These complex algorithms are useful for uncovering highly non-linear relationships between the predictors and the outcomes (I think of image recognition using pixels as inputs as an example of this). However, I don't think these complex non-linear relationships exist in our specific application. For that reason, linear regression (which, of course, is still quite flexible due to the possibility of including higher-order terms) appears to be sufficient for our purposes.

GOLF, PREDICTIVE MODELLING, STATISTICS