CS229 Term Project
How's Your Golf Game?

Steve Goldman, Drew Willcoxon

December 16, 2005

Introduction

Fore!!

The game of golf has tortured the minds of leisurely and competitive enthusiasts for over a century. How could the task of striking a ball so that it ends up in a hole be so difficult? After all, the ball doesn't move until you hit it. Despite its frustrating nature, millions toil endlessly to improve their game. Unfortunately, due to the pesky demands of life like work and family, for most, available practice time is scarce. This project seeks to identify the aspects of golf that most directly impact the bottom line: a player's score.

We collected data from *www.pgatour.com* about 80 of the top 200 players on the tour. We used cross-validation and feature selection techniques to construct a regression model that estimates a player's average score based on the attributes of his game. Next, we performed experiments that identify which parts of the game contribute to scoring. Finally, we designed a search system that partitions his finite practice time into the activities that will improve his score the most, given the state of his game and his estimates for the return on investing time in each practice area.

Data

A Fine Walk Spoiled

There are many aspects of golf that can be captured by statistics. Primarily, we are interested in how many strokes a player takes in a round. We use other measures, like how far or how accurately a player hits the ball, to build a model that outputs scoring average. We chose the following eight attributes to consider in our model.

Attributes

Driving Distance The average distance in yards that a player hits the ball from the tee when he is trying to hit is as far as he can.

Driving Accuracy The proportion of the time a player's drive ends up in the fairway.

Greens in Regulation (GIR) The proportion of the time that a player's ball ends up on the putting green in regulation, where regulation is one stroke on a par 3 hole, two strokes on a par 4, and three strokes on a par 5.

Putts per Hole (GIR) The average number of putts a player takes on a hole in which he has hit the green in regulation.

Sand Saves The proportion of time a player makes par or better on a hole in which he was in a sand trap.

Putts per Round The average number of putts per 18 holes.

Scrambling The proportion of time a player makes par or better when he does not hit the green in regulation.

3-putt Avoidance The proportion of time a player requires three putts on a hole.

Why these attributes?

The attributes we selected are well-known to golfers. If we tell a golfer to improve on one of these, the golfer will know what to practice. For instance, to improve scrambling, golfers know to practice chipping, pitching, trouble shots, and medium length putting.

Why only professionals? Why only men?

Originally, we intended the model to describe the population of all golfers. We used data about professionals because that is the only data available; to our knowledge, there is no source of rich data about the attributes of amateur golfers. Early on, however, we observed nonsense output for test inputs about amateurs (attributes that describe our own golf games). We determined this was due to the lamppost effect. Thus, we restricted our population to professionals.

As for gender, the model is based on statistics about golfers who mostly played the same golf courses. Since professional women and men play different types of golf courses (courses on the LPGA tour tend to be shorter and narrower), women's attributes are rewarded differently than men's. We could construct a model in exactly the same way using data about women golfers. In fact, this is a possible project extension.

Model Construction and Testing

Must You Breathe While I'm Putting?

As a motivation for creating the model, we computed the correlation between each individual attribute and score.

Attribute	Correlation
Driving Distance	-0.3439
Driving Accuracy	-0.0136
Greens in Regulation	-0.6732
Putting (GIR)	0.2579
Sand Saves	-0.1412
Putts per Round	-0.0704
Scrambling	-0.4114
3-putt Avoidance	0.1155

We see that no single attribute is a good indicator of final score. The best is GIR with -0.6732. However, statistically, this is not a strong correlation. Therefore, we chose to construct a regression model that combines all the attributes to make quality predictions.

First we scaled the input attributes to have the same means and standard deviations. Then we applied linear regression. We used the closed form equation for the parameter vector that minimizes the least squares error in linear regression.

$$\theta = (X^T X)^{-1} X^T Y$$

where X is the feature matrix and Y is the output vector of the training examples. For testing, we used leave one out cross-validation (LOOCV). This was effective because the training set was small enough so that LOOCV was not too time consuming. As a result, we avoided selecting models with high variance that fit the training set extremely well at the expense of miscalculating future input.

Feature Selection

Grip it and Rip it!

Given our eight attributes, first we defined a feature as a product of attributes. By this definition, there were $n = 2^8 - 1 = 255$ features. We knew we probably didn't want to use all of them in our model, so we used a forward search to choose which features to use. Our algorithm was:

$F = \emptyset$
for $i = 1$ to n
construct model for $F \cup f_i$
repeat
$F = F \cup f$

where the model with f gave the best cross-validation error. We repeated this algorithm, adding features to our feature set one at a time, until the cross-validation error stopped decreasing. The model we chose had 15 features and a cross-validation error of 0.165137.

Trying to get a more tightly fitting model, we redefined a feature as a product of attributes, where each attribute is either linear or quadratic in the product. By this definition, there were $n = 3^8 - 1 = 6560$ features. Again, we applied the forward search to choose features for the model. The result was a model with 16 features and a cross-validation error of 0.131037.

But why stop there? We again redefined a feature, this time as a product of attributes, where each is either linear, quadratic, or cubic in the product. After forward search, we chose a model with 16 features and a cross-validation error of 0.117694.

We settled with this model. See figure 1 for a comparison between the three feature definitions. We stopped here since the $4^8 - 1 = 65535$ features already pushed our computation capabilities. Allowing quartic attributes in features would have been computationally prohibitive. We also considered allowing other functions of attributes in our attribute products, such as log, exp, and fractional exponents; however, deciding how to grow the size of the feature set seemed arbitrary. Furthermore, we expected that even larger feature set sizes would yield diminishing returns in terms of lowering cross-validation error, and would probably lead to over fitting and high variance. The error in the final model was acceptable, so we decided to use the model.

Figure 2 shows the relationship between training error and cross-validation error during the forward search. We expect that as the forward search continues and adds more features to the feature set, the training error will approach zero. However, we stop at 16 features because cross-validation error is a better indicator for error on future data.

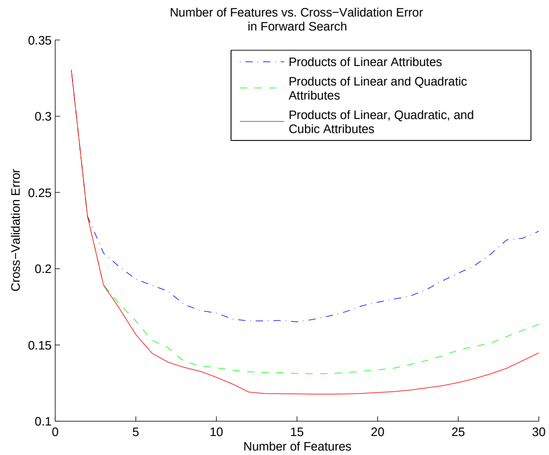

Figure 1: Cross-validation error for feature sets of increasing size, as constructed by the forward search algorithm, for the three definitions of feature.

Experiment

Drive For Show and Putt For Dough?

Our goal at the outset was to determine which attributes influence scoring the most. We created an accurate model based on these attributes, but since the model mixes and matches attributes, we cannot directly detect which are the most important. Naively, we can calculate the number of features each attribute is a factor of. See figure 3. Driving distance is a factor of the most attributes at 11. The two driving attributes appear 18 times in the model, as do the three putting attributes. However, these observations don't tell us anything meaningful about the attributes' influence in the model.

To glean information about influence from the model, we devised an experiment that shows what influence changing one attribute has on the model while leaving the rest the same. First we created 10 test golfers, giving each random values for attributes, all within .5 of the mean. Then, one at-

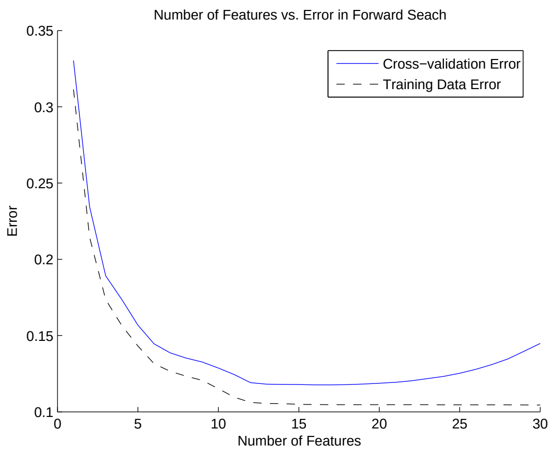

Figure 2: Training and cross-validation error for the final model.

tribute at a time, we changed the value of each test example by some constant ϵ. Then we compared the difference between the output of the model for each example with and without the change. Finally, we averaged this difference over the test set. The results show how changing each attribute affects the output of the model. Figure 4 shows the effects of changing each attribute between -2 and 2 standard deviations. We chose these bounds because anything outside of them would represent an unreasonable change in a player's game. (Note, since we standardized the training data to have the same means and standard deviations, changes of ϵ affect each attribute equally.)

Results

The attribute that jumps out is GIR. This result is expected, because when a player hits a green in regulation, he almost always makes par or better. So, the more GIRs, the more pars and birdies, and the fewer strokes in the round.

Of the putting attributes, the most influential, according to the graph, is putts per round. There is

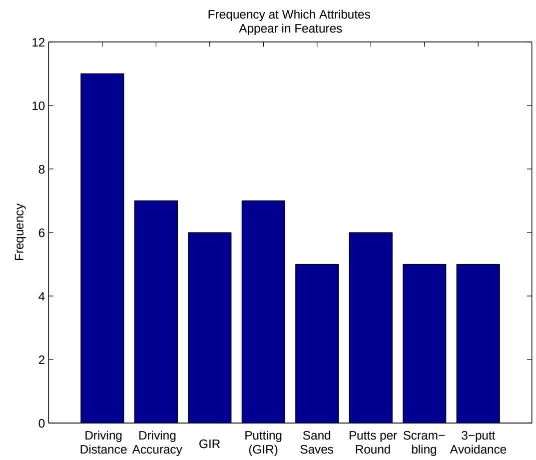

Figure 3: The number of features each attribute is a factor of.

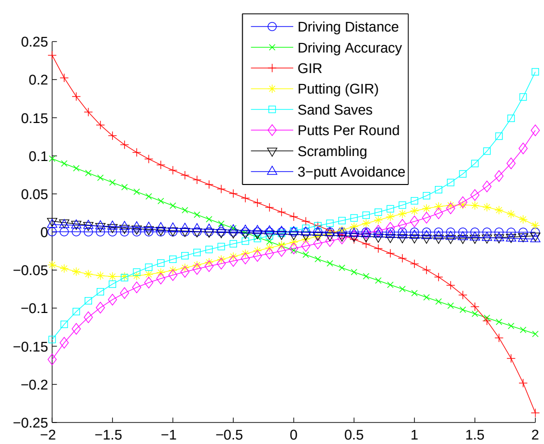

Figure 4: The effects of changing each attribute by different amounts.

a clear correlation between putts and score. This makes sense, since this attribute translates directly to score; each putt contributes one stroke to the score. Putting (GIR) has a similar effect as putts per round, yet to a smaller degree. Apparently, hitting the green in regulation is more important than sinking the putts afterward. 3-putt avoidance, surprisingly, has little effect on the score. There is a positive correlation, but it is so insignificant, it is difficult to detect.

As for driving attributes, driving accuracy has a strong negative correlation with score. We expect this result since any hacker[1] will tell you it is easier to score well when you drive the ball in the fairway.

The following result is somewhat shocking. In our model, driving distance has almost no effect on average score. The graph shows driving distance almost exactly on the x-axis.

Finally, we look at the last two attributes. Scrambling is similar to 3-putt avoidance; it has little effect. The graph for sand saves, however, reveals a problem with the model. Obviously, improving efficiency from the sand should not cause score to increase. Yet, the graph indicates that it does. This could be a problem of over fitting. This is an open problem in our research group.

Analysis and Conclusion

So, is the old adage is true? In golf, do we really drive for show and putt for dough? Indeed it is, to a certain extent. Perhaps the results of this project can help refine the aphorism: drive it long for show and putt for dough and drive it straight for dough, too. Our experiment shows that whereas driving for distance has little effect, driving accuracy has a strong influence on score. This indicates that the tee shot is important in golf, but players should focus on hitting it straighter, rather than longer.

[1] Do not confuse this with a computer hacker. In golf, a hacker is someone who is perpetually playing from the high grass. He must "hack" his way back to the fairway.

Future Work

Heading to the Range

As an extension to this project, we designed but did not implement a search strategy for determining an optimal practice schedule. Suppose a player only has a finite amount of time, T hours, to practice this month, which we divide into n equal length partitions. Further, suppose there are k practice activities into which he can divide his time; i.e. driving range, putting green, weight room, etc. Finally, suppose he knows *apriori* the effect that practicing one unit of time has on all his attributes for each activity. Then we can construct a tree whose depth is k, where each level corresponds to a practice activity, and each branch corresponds to the amount of time spent practicing the child activity. Each leaf node is a practice schedule, and from it we can compute the resulting attribute vector. Then we run all these attribute vectors through the model, and select the one with the lowest output. This corresponds to the ideal practice schedule.

However, the design is ineffective for modest size k and n, since the size of the tree is on the order of k^n. Therefore, we must limit both the number of practice activities and the number of partitions of T. Yet, for small values, such as $k = 4$ and $n = 10$, the design would work.

An extension to this design would be to model practicing as a probabilistic event. Any golfer will tell you that not all practice is good practice. That is, practicing chipping usually improves chipping, but sometimes it makes it worse. Then we can view the problem as an MDP. However, there is still a problem with exponential number of states.