Predicting Golf Score Using Hidden Markov models

William 'Scott' Hanna

***Abstract*— This paper applies the concept of Hidden Markov models and the Viterbi algorithm in order to predict the total score of a round of golf. The idea was to see if given the mood of a William & Mary varsity golfer the Viterbi algorithm could accurately predict the total score of the golfer after the completion of the round. In total, 20 rounds of 18 hole data was collected, keeping track of the score and mood of each respective player after every hole. The mood of the golfer was a very good indicator of their final score, but not necessarily their score from hole-to-hole.**

I. INTRODUCTION

In this paper we will explore the topics of Hidden Markov models and the Viterbi algorithm. The 'Background' section of this paper will be split into various parts. The Hidden Markov models section describes the necessary mathematics to understand the Viterbi algorithm. The terminology section describes the golf terminology that is relevant for the experiment. After the necessary background is established the Viterbi algorithm is described in detail. In the experiment, we test to see if the mood of a golfer after the completion of each hole is an accurate indicator of their score on that hole. Each hole can be considered a separate event, the chain of the holes can be thought of as a Markov chain. The Viterbi algorithm takes the observed mood of the player and outputs what score is the most likely on that hole.

II. TERMINOLOGY

To understand the experiment that is the focus of this paper some golf-related terms must be defined.

- A round of golf is comprised of 18 different holes. In our experiment we consider each hole as a separate event
- The score relative to par on a single hole is the difference between the recorded score and the par. For example, a score of 5 on a par 5 has a score relative to par of 0 while a score of 4 on a par 5 has a score relative to par of -1.
- The score relative to par of a round is the summation of every score relative to par on a hole.

III. HIDDEN MARKOV MODELS

A Markov model is a sequence of events in which the next event is only dependent on the current event. If the sequence of events is discrete we can describe the chain as:

$$p(X_1)\prod_{i=2}^{n} p(X_i|X_{i-1})$$

Here we can see that this a chain of conditional probabilities in which an event X_i is only dependent on X_{i-1}. Markov models have multiple applications including language processing models and browser page rank algorithms [1].

A hidden Markov model consists of a discrete Markov model whose events are unknown and a sequence of observations whose likelihood depends on the events in the Markov model. We can describe HMMs as:

$$(p(X_1)\prod_{i=2}^{n} p(X_i|X_{i-1}))(\prod_{i=1}^{n} p(Z_i|X_{i-1}))$$

Where (X_n) is the sequence of hidden events and (Z_n) is the sequence of dependent observations [2].

IV. VITERBI ALGORITHM

The Viterbi algorithm finds the sequence of states that is most likely to result in the given sequence of observations. The Viterbi algorithm requires five inputs [3]:

- A set of observations (O)
- A set of initial probabilities (J)
- A transition matrix which contains the probabilities that one state transition to another (T)
- An emission matrix which contains the probabilities that one state corresponds to every possible observation (E)

The overarching idea of the Viterbi algorithm is to find what state at the time of a certain observation is most likely. There are three steps to the Viterbi algorithm [4]:

1. The algorithm starts by finding the most likely initial state by computing

$$P_{s_k}(1) = P(s_k|s^j)P(o_1|s_k)$$

$P(o_1|s_k)$ is the probability of the first observation given each state $s \in S$ which is given by E. $P(s_k|s^j)$ is the probability given by J that the given state is the start state for each $s \in S$. Looking at all possible combinations we can find the most likely starting state.

2. The next step is finding the most likely state for each observation given by O. The key point is that we must take into account what happened in the event directly before the current event. We calculate the likelihood of each possible state for observation i using the following formula:

$$P_{s_k}(i) = P(s_k|s^j)P(o_i|s_k)P_{s_k}(i-1)$$

The main difference between this step and the step above is that we now must take into account the previous state, this is when we apply the transition matrix. It should be noted that we will choose the max $P_{s_k}(i)$ because we will calculate it for each possible state. We will then add the most likely state to the chain. Another note is that the number of chains calculated is equal to the number of possible starting states.

3. Once we finish calculating all the chains we examine the likelihood of each chain and choose the chain has the largest probability of happening. In this way it is similar to the Näive Bayes algorithm. The final result is the chain of states which had the maximal probability.

V. Viterbi Experiment

A. Set up

The idea behind this experiment is to see if given a William and Mary golfer's mood after each hole, his total score could be predicted with accuracy. The data contains 20 18-hole rounds. Although 20 data points may not seem like a lot, in reality there are 360 data points as the holes are what the various parameters are based upon. The data was collected by observing various players on the W&M golf team complete a round of golf, noting their score and mood after each hole was completed. The following section describes how each parameter was found:

- The possible states are: [-2,-1,0,1,2]. This represents how the golfer did on each hole relative to par. So if the golfer is playing a par 4 and makes a score of 4 the corresponding state is 0, if they make a score of 5 the state is 1. In reality golfers make scores higher than 2 over par, but at this level of play it happens so rarely that I decided to cap it so that outliers would not affect the data.
- The set of initial probabilities is computed by seeing how many times each state was the first state in a round of golf.
- For the transition matrix (figure 1) each state must be examined separately. Let us examine state 0 for example. Every time 0 appears in the training data we note the score of the next hole. So in our data the score 0 was made 154 times in the first 17 holes of a round. 0's on the 18th hole are not included because there is no next hole. After a score of 0 on the previous hole a score of -1 was made 40 times. That means the probability of making a score of -1 after a score of 0 will be $\frac{40}{154}$ in the transition matrix. This process will be completed five times for 0, once for each possible state.
- For the emission matrix (figure 2) each state must be examined separately. Again let us examine state 0 as an example. It should first be noted there are five possible observations (meh, good, great, bad, sarcastic). Every time a 0 is made we check the corresponding observation of the mood of the golfer. First of all, there are 166 0's made in the training data. The 'meh' observation was made 65 times when a 0 was made. So in the emission matrix, for the state of 0, the probability of 'meh' is $\frac{65}{166}$.

{0: {0: 0.565 , 1: 0.169, 2: 0.006, -1: 0.256, -2: 0},
1: {0: 0.481, 1: 0.25, 2: 0.058, -1: 0.212, -2: 0 },
2: {0: 1, 1: 0, 2: 0, -1: 0, -2: 0 }
-1: {0: 0.672, 1: 0.180, 2: 0, -1:0.131 , -2: 0.016},
-2: {0: 0, 1: 0, 2: 0, -1: 1, -2: 0 }}

Fig. 1. Transition Matrix

{0:{"meh": 0.392, "sarcastic": 0.108, "bad": 0.120, "good": 0.337, "great": 0.042},
1:{"meh": 0.364, "sarcastic" 0:.145, "bad": 0.436, "good": 0.055, "great": 0},
2:{"meh" :0,"sarcastic": 0,"bad": 1,"good": 0,"great": 0},
-1:{"meh": 0.311,"sarcastic": 0,"bad": 0.016, "good": 0.459,"great": 0.213},
-2:{"meh": 0,"sarcastic": 0,"bad": 0, "good": 0,"great": 1}}

Fig. 2. Emission Matrix

B. Results

Now let us examine how well our parameters predict a full round. There are 3 testing rounds, here are the results:

1) Real score of -1, guess of -1 (figure 8)
2) Real score of -3, guess of -4 (figure 9)
3) Real score of 2, guess of 3 (figure 10)

At first it seems as though the model is very accurate, and when only the total round is considered it is accurate. However we see a different story when we examine the hole by hole predictions that the model makes.

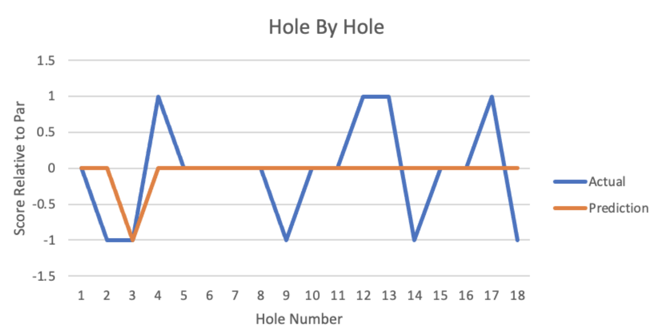

Fig. 3. Test 1 - hole to hole

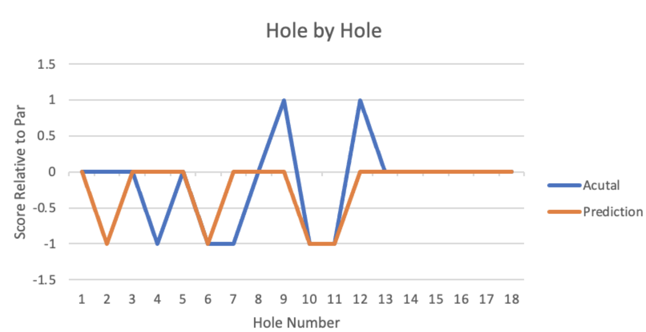

Fig. 4. Test 2- hole to hole

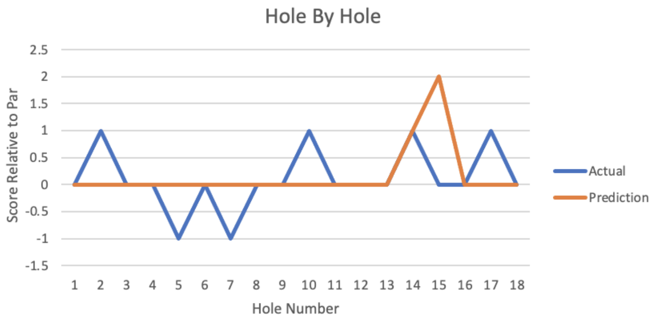

Fig. 5. Test 3 - hole to hole

As seen in figures 3,4, and 5, the accuracy from hole-to-hole is not very good, roughly 67%. This is for a couple of reasons. The first is that the golfers that were studied are humans, and have a memory going back further than 15 minutes. The key part of a Markov chain is that the next result only depends on the current state. The mood of a golfer is not dependent only on how they played the current hole. Instead it is dependent on how their overall round is going in general, with some recency bias. The second reason is the nature of the Viterbi algorithm. In our training data there were 166 pars made out of 288 holes played, that's about 58%. The algorithm predicted 47 pars in 54 holes, about 87%. Since the Viterbi algorithm picks the most likely result every time, it is bound to choose the result of par too much as it is often the most likely. In order to address this problem changes to the transition matrix must be made. The results of these changes are discussed in the Laplace Smoothing subsection.

Next, lets put how good the results of the total score experiment are into perspective. In our training data there were total scores as low as -9 and as high as +12. That is a wide dispersion, but most of the score are in the middle. The model's total score predictions were off by .67 shots per round, an accuracy of just under 97% when considering the possible range of scores. So if the hole to hole predictions aren't very good, then why are the overall score predictions good? As stated above, the model over predicts the score of par. In reality, W&M golfers make a lot more birdies and bogeys than the model predicts. The total number of bogeys, 55, and the total number of birdies, 61, is relatively close. So, when the model predicts a par instead of a birdie or bogey it tends to self correct later on. For example, let us say the model predicts a score of par when the player actually made a birdie on hole number 4. Later on, the model is likely to predict a score of par on a hole where the player actually made bogey so it cancels out.

The goal of this experiment was to predict the total score of the player, not the hole by hole scores. So, the model is generally successful.

C. Laplace Smoothing

One of the issues with our model could be that certain outcomes are given a 0% of happening. Although the amount of data collected was extensive, as previously discussed, it did not take into account every possible scenario. In order to address this Laplace smoothing will be applied to the transition matrix.

```
{ 0: {0: 0.564, 1: 0.168, 2: 0.006, -1: 0.256, -2: 0.002},
  1: {0: 0.481, 1: 0.240, 2: 0.057, -1: 0.211, -2: 0.012},
  2: {0: 0.555, 1: 0.166, 2: 0.028, -1: 0.222, -2: 0.028},
 -1: {0: 0.670, 1: 0.180, 2: 0.007, -1: 0.130, -2: 0.016},
 -2: {0: 0.222, 1: 0.166, 2: 0.028 ,-1: 0.555, -2: 0.028}}
```

Fig. 6. Transition Matrix after Laplace

The new values, were calculated based on how often that score happens and then normalized so the total probability is equal to 1.

The results after Laplace smoothing were only slightly better. The accuracy hole to hole was slightly better, around 70%. Interestingly, this actually resulted in better overall testing. The third test which originally had a guess of 3 despite having a real score of 2, was predicted correctly this time (figure 11). Even though there was slight improvement, the Laplace smoothing was generally inconsequential. The improvement were modest, and 70% is still not amazing.

D. Limitations

Here the limitations of this model should be mentioned. First, the model will only work for players close to the skill of a W&M varsity golfer. This isn't surprising, as the W&M varsity golfers are what the model is trained on. Second, and more interestingly, is that the model only works on certain golf courses. The model will work fairly well for golf courses that are around the same difficulty as the golf courses the W&M players competed on in the training data. However, when the golf course increases or decreases substantially in difficulty the predictions become less accurate. Here is a round at Royal New Kent Golf Club:

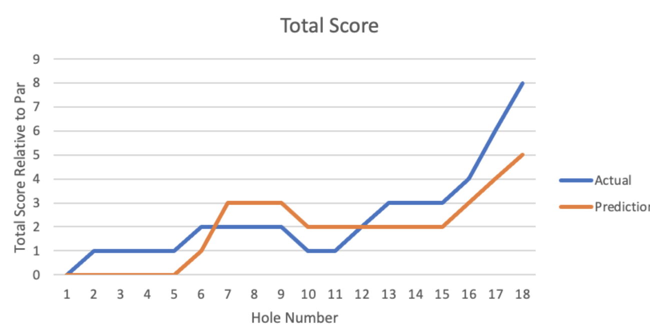

Fig. 7. Royal New Kent Total

The total score in this test is off by three shots, an accuracy of 86%. The accuracy is off because Royal New Kent is a much more difficult golf course than the courses in the training data. The golfer is happier with a score of +8 at Royal New Kent, than he would be at one of the training golf courses. Since the golfer is happier, the model predicts a lower score.

VI. Miscellaneous Figures

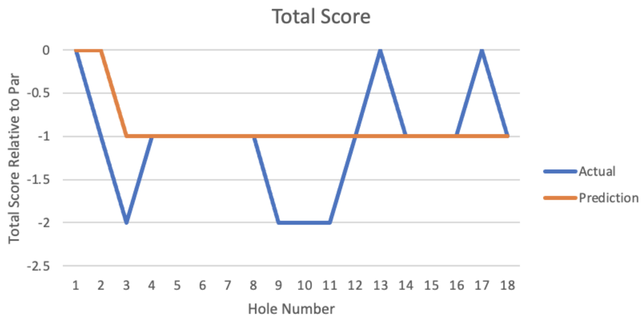

Fig. 8. Total score test 1

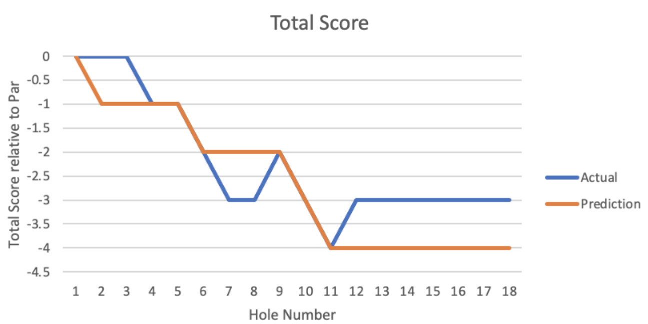

Fig. 9. Total score test 2

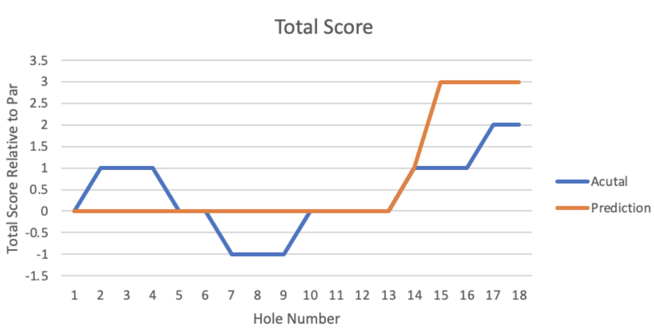

Fig. 10. Total score test 3

Acknowledgment

I need to thank my teammates for allowing me to collect data on them while they played, I would not have been able to complete this paper and research without them.

References

[1] K. P. Murphy, Machine learning : a probabilistic perspective. Cambridge (Ma): Mit Press, 2012, pp. 591.

[2] K. P. Murphy, Machine learning : a probabilistic perspective. Cambridge (Ma): Mit Press, 2012, pp. 606.

[3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989, doi: https://doi.org/10.1109/5.18626.

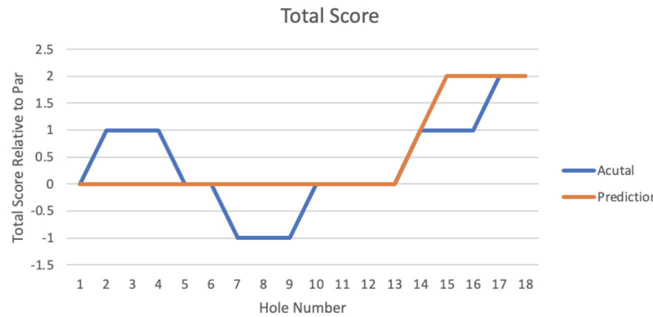

Fig. 11. Total score test 3 after Laplace Smoothing

[4] P. Butler, "Intro to the Viterbi algorithm," MLearning.ai, Mar. 02, 2021. https://medium.com/mlearning-ai/intro-to-the-Viterbi-algorithm-8f41c3f43cf3: :text=The%20Viterbi%20algorithm%20is%20composed %20of%20three%20steps.&text=We%20first%20create%20a%20start (accessed Dec. 13, 2023).