CS221 Progress Report

TJ Melanson, Matt Millett **Date:** November 11, 2015

---

1 Race Predictor Model

Racing ability, at its core, consists of two factors: speed and endurance. Speed defined here is how fast a runner is capable of running. We propose a correlation between speed, endurance, distance, and time:

$t = s(d^E)$

where d is distance, t is time, s is speed, and E is endurance.

A runners speed improves in small amounts. Additionally, the importance of speed decreases with distance: it is crucial for a short race, but almost inconsequential for a long race. Therefore, it is represented as a linear modifier, s.

Endurance, however, can vary greatly between person to person, and matters much more for long races than short races. It is then treated as an exponent E, where $1 < E < 2$. Think of the s value as a runner's full potential power, and E the atrophy of that ability over the distance run. Peter Riegel, who is best known for hypothesizing the exponential model. approximated the value of E as 0.06. However, this value varies depending on experience, age, and other factors, and often is not accurate for the majority of cases.

2 Regression on the Model

Because the Riegel formula is of the form $y = Ax^b$, we decided to use two forms of regression: one to find the value of s and one to find the value of E. The value of s is a simple linear regression:

$s = s - \eta \ \nabla_s Loss(s, E, d, t)$

The Loss function, because the value of t can be very large, proved to diverge when calculated as squared loss. However, absolute value regression proved to converge as long as η remains small.

Pure linear regression cannot work for E, however, as a change in E does not correspond to a linear change in t. Taking the log function, however, brings the E outside of the formula, like so:

$\phi(sd^E) = E\log(d) + \log(s)$

where $\log(s)$ is treated as a sort of t_0 initial value. The expected value is then $\log(t)$, and linear regression can be done on the new value:

$E = E - \eta\ \nabla_s Loss'(s, E, d, \log(r))$

Where the Loss function is the squared-loss function of $\phi(sd^E)$ and $\log(t)$.

3 Weighted (Time-based) Regression

The regressions calculated above work well for a single snapshot of a runner's career. However, over several seasons, a runner can improve and adapt, or, inversely, get out of shape from not running. Because of this, we needed a way to take into account the time elapsed between runs. A race completed a week ago is a strong indicator of fitness, while a race completed 5 years ago is much less relevant.

To adjust for this, we considered assigning weights to the Loss function so that the cost is greater the more recent the race. However, because the data between races might not change significantly, we decided to lose this model in favor of using a Hidden Markov Model, where the hidden variables are fitnesses over time.

4 Creating a (s,E) Distribution

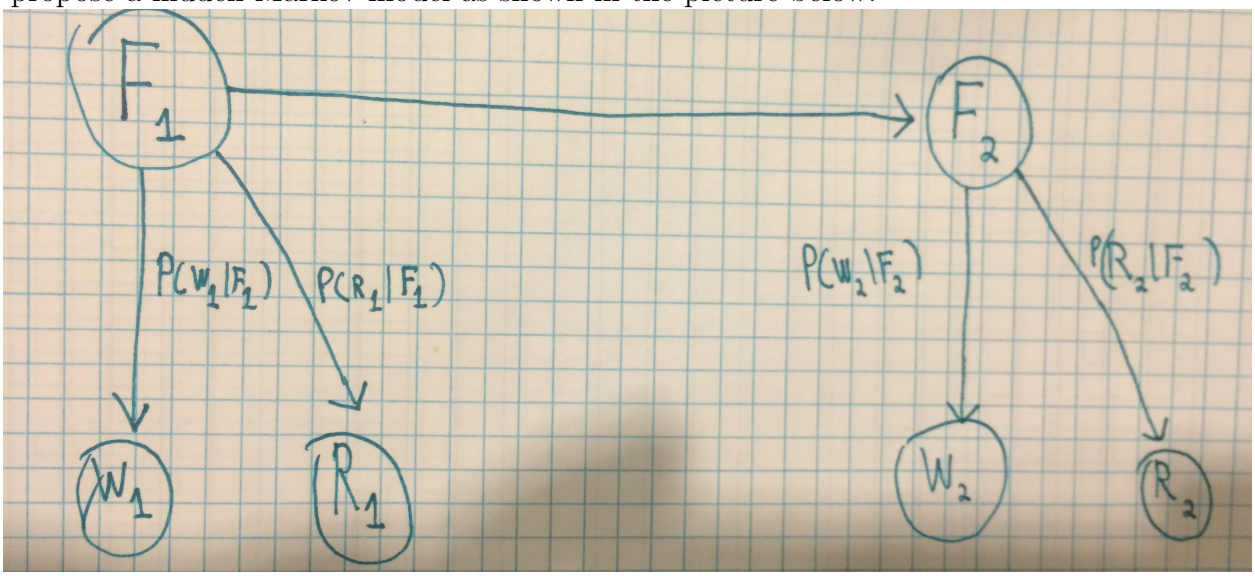
While regression for the value of $t = s(d^E)$ gives a somewhat accurate answer, most runners can give only 4 or 5 race times at the most. This makes an accurate regression model nearly impossible; however, because the values of E and s vary between people, we cannot use multiple datasets to calculate these values. Therefore, we hypothesized using Bayesian inference to determine (s, E) from a given (d, t).

The major issue for finding the distribution this way is that the function needs to cover all values of d and t – i.e. the probability distributions must be continuous with respect to d and t. To solve this, we use an overall regression function for the value of (s, E) given a value of (d, t), and adjust it with the actual values of (d, t) and (s, E). Because s, E, d, and t are all interdependent, if three of the variables are found, then the value of the fourth is

deterministic. This is why, if we know the values of d and t, (s, E) can be treated like a single variable: finding the value of s can lead to E, and vice versa.

5 Hidden Markov Model

We propose a hidden Markov model as shown in the picture below.



F_t for a time period t is the fitness level of an athlete during time period t, and is parameterized by the endurance value E_t and the speed value s. W_t and R_t are, respectively, the set of workouts and the set of races done in the time period t. Note that workouts and races do not directly depend on each other, but knowing workout data *does* affect fitness, which in turn affects race predictions. By running the power regression discussed above on the data to find values for s and E, we find the expected values of s and E, with some standard deviation. We also make the assumption that peoples' fitness levels are normally distributed based on these signals, where the mean is the expected value of (s, E) from the power regression, and the standard deviation also comes from there. This gives us $P(F_t; s_t, E_t|R_1)$. We're still working on getting workout data and our model can function without it, so we haven't decided how to figure out values for s and E from workouts yet, because they may not be relevant.

Our model allows us to aggregate the runner's entire race and workout history to say what he/she could have run in the past, or what he/she could probably run now.

This general model allows us to individualize race prediction and output the probability

a runner will run a certain time.

Lastly, we need to find the transition probability for fitness states. This is where having a large data set from races comes in. The transition probability from one fitness state to another can be determined using all our data sets, so we will be able to see how much a runner usually transitions within a time period. For specific groups of athletes, we can feed our model more specific transition probabilities (e.g. an elite athlete is less likely to have a large transition in s and E whereas someone in a couch-to-5k program will be much more likely to have significant changes in these values.

We haven't been able to collect significant amounts of workout data yet, only race data, so we will figure out exactly how to run the power regression on workout data if we get it. If we do get workout data, we will find s and E to be the weighted sums of the s and E values returned from regressions on W and R, using $0.2s_w + 0.8s_r = s_{tot}$ and $0.2E_w + 0.8E_r = E_{tot}$. While this is inexact, we may be able to refine these coefficients later.

A hidden Markov model should do better than a direct regression because it more directly shows the ability of runners to grow over time and allows more for the randomness in workouts and races. A weighted regression does not incorporate these factors as much, which is why the HMM might perform better.

6 Additional notes on data

We found a large repository of race times that correspond to over 100 individuals over the past 3 years. We are planning to use this data for the final implementation, but we ran our test on the smaller data set we used from the proposal.

7 Preliminary results from the power regression

Edward Cheserek s: 0.10472214522107559, E: 1.0490867117157918
Error 0.4
Ben Saarel s: 0.11461022235915472, E: 1.046131712379568
Error 0.354929577465
Shaun Thompson s: 0.09299180794561976, E: 1.0442828593347562
Error 0.472675656494
Patrick Tiernan s: 0.10068280994874888, E: 1.0337397048064758

Error 0.416666666667
Sean McGorty s: 0.15773748552450728, E: 1.0357669046855145
Error 0.0908435472242
Malachy Schrobilgen s: 0.14549197609740708, E: 1.0327720654691575
Error 0.178091872792
Thomas Curtin s: 0.11242849139671121, E: 1.0481281777635472
Error 0.356938483548
Jonathan Green s: 0.10497683199906799, E: 1.0444530962668126
Error 0.404119318182
Marc Scott s: 0.10747174080550748, E: 1.0385030122795302
Error 0.390780141844
Joe Rosa s: 0.06709966144810668, E: 1.0572726968800574
Error 0.616045845272
Pierce Murphy s: 0.13224001166726063, E: 1.0343959071630462
Error 0.247150997151
Anthony Rotich s: 0.10747174080550748, E: 1.0385030122795302
Error 0.377986965967
Martin Hehir s: 0.10920267677134733, E: 1.0544709920465836
Error 0.374193548387
Grant Fisher s: 0.07463655155623343, E: 1.0321952525053435
Error 0.577494692144
Ammar Moussa s: 0.1267217407377259, E: 1.0417480403832409
Error 0.283592644979
Justyn Knight s: 0.10464405321139912, E: 1.0389037330966806
Error 0.406382978723
Minimum total error: 0.0908435472242