Thesis Manuscript

Andrew Glover

2024-03-16

Introduction

One of the major goals of sports science is to maximize athletic performance. A key aspect of this optimization is the training regimen that coaches prescribe an athlete. This training regimen involves training load, such as the exercises performed, and the number of sets and repetitions performed, and the recovery program, including diet and sleep. For some activities, such as running, swimming, power lifting, and shot-put, training load and the performance output has a measurable external component. The quantity Training Load captures the effect of the training on the body, through such methods as measuring VO2 max (the amount of oxygen that is avaiable for use by the body; the higher the V02 max, the more athletically fit the athlete is) for endurance training or a method direct quantification of the training load – counting weight, reps (how many repetitions to lift the weight), and sets (how many times to perform the given number of reps) – for resistance based exercises, like power lifting. It logical and correct to expect that sports where training load can be quantified have a quantifiable performance output. The performance for running and swimming can be quantified by the meet time, powerlifting by one rep max (how much weight you can lift one time), and shot put by distance thrown. When modeling the relationship between training load and athletic performance, it is important to choose a model that captures the physiological systems that underly recovery, allowing sports science researchers to use the model to be able to prescribe or inform a training regimen. This approach cannot hope to perfectly be able to capture everything, as there are other important factors to recovery, but it can capture most of the relationship, and almost all in some situations, like elite athletes. The gold standard for this approach is the Banister Fitness Fatigue model.

Literature Review

The Banister model was introduced in 1975 by Banister et al to model athletic performance from training load (Banister et al. 1975). However, this model didn't account for an observed initial dip in the performance, so an additional term accounting for negative effects was added, giving rise to the Banister Fitness-Fatigue model. (Banister 1976). It has been used over the years to when the performance ability at a specific time can be easily quantified, such as swimming, cycling, power lifting, running, and the hammer throw (Mujika et al. 1996; Thomas, Mujika, and Busso 2008; Busso, Candau, and Lacour 1994; Busso et al. 1997). The Banister FF model does have it's limitations. Model stability, the model parameters not being intrepretable, predictive accuracy, and ill-conditioning have been reported (Hellard et al. 2006; Ludwig et al. 2019). Additionally, provide a good explanation of the limitations of this model (Vermeire et al. 2022). Previously, there has been a lack of publicly available computational resources for estimating and applying the model. However, Clarke and Skiba give a great introduction and overview of this model and apply it using Excel(Clarke and Skiba 2013). Also, this blog post by Andreas K. Winther applies the model in R, and was helpful in creating some of the code utilized for this project (Winther 2019).

Methods

The Banister Fitness-Fatigue Model

The Banister Fitness-Fatigue model (BFFM) models athletic performance as a function of training load. Roughly, it says that performance can be modeled as the sum of the ability of the athlete before training begins, the positive effects of training (PTE), and the negative effects of training (NTE). Formally, it is defined as,

$$P(n) = p_0 + \overbrace{k_1 \sum_{i=0}^{n-1} e^{-(n-i)/\tau_1} w(i)}^{\text{PTE}} - \overbrace{k_2 \sum_{i=0}^{n-1} e^{-(n-i)/\tau_2} w(i)}^{\text{NTE}}$$

where p_0 is the initial performance, $w(i)$ is the training load on day i. $k_1,\ k_2,\ \tau_1$, and τ_2 are real, positive constants. In this model we use the convention that p_0 occurs on day 0. For computational ease, $P(n)$ can be written in terms of recursion equations:

$$p(n) = p_0 + k_1 \cdot g(n) - k_2 \cdot h(n)$$

where $g(0) = h(0) = 0$ and

$$g(i) = e^{-1/\tau_1}\Big(g(i-1) + w(i)\Big), \quad h(i) = e^{-1/\tau_2}\Big(g(i-1) + w(i)\Big)$$

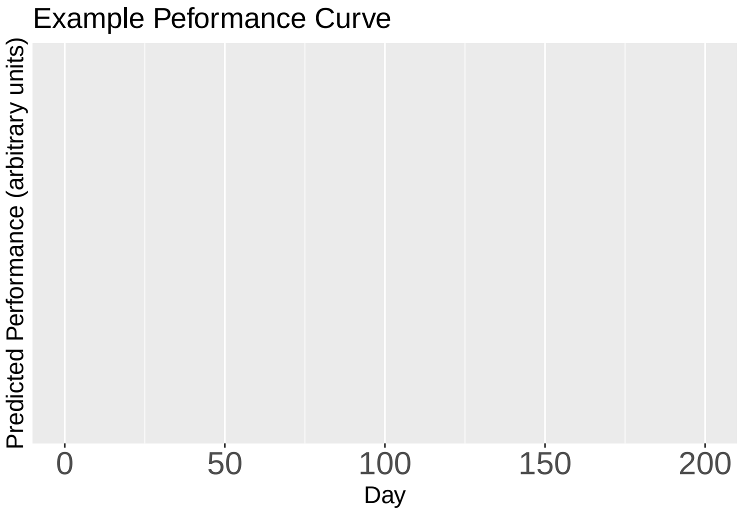
These equations have the equivalent formulation:

$$p(n) = p_0 + P(n) - N(n)$$

where $P(0) = N(0) = 0$ and

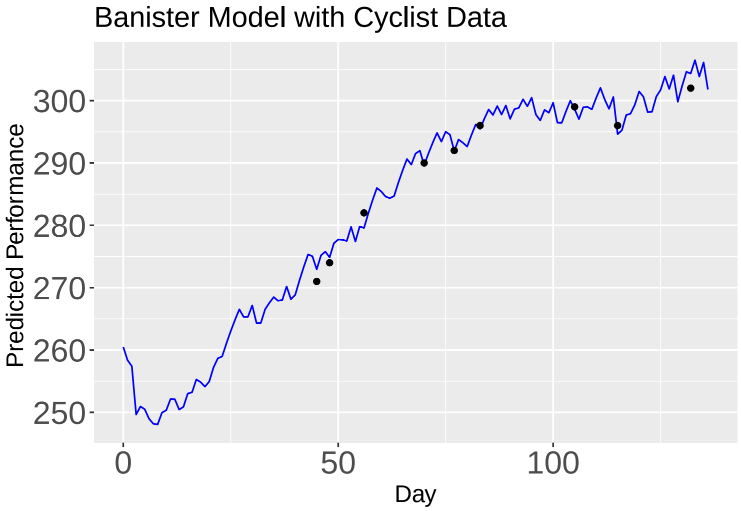
$$g(i) = e^{-1/\tau_1}\Big(P(i-1) + k_1 \cdot w(i)\Big), \quad h(i) = e^{-1/\tau_2}\Big(N(i-1) + k_2 \cdot w(i)\Big)$$

It is instructive to have an intuitve idea as to how this model works. Below is a graph of the BFFM with parameters $(p_0, k_1, k_2, \tau_2, \tau_2) = (500, 1, 2, 30, 10)$, with a constant training load of 100, i.e. $w(i) = 100$ for all i, for 200 days.



This captures a few aspects of the performance-training relationship: an initial negative response to the training, an eventual recovery and a benefit to the training, a plateu in the long run.

This is what the Time-Invariant model looks like when it is applied to a real dataset



This model displays many of the same characteristics as the previous model.

Intuitive Explanation of the Banister FF Model

Lets just look at the sum. TE here stands for either positive training effects or negative training effects. We have

$$\text{TE}(n) \propto e^{-1/\tau}w(n-1) + e^{-2/\tau}w(n-2) + \cdots + e^{-n/\tau}w(0)$$

or, written differently,

$$\text{TE}(n) \propto (\lambda)^1 w(n-1) + (\lambda)^2 w(n-2) + \cdots + (\lambda)^n w(0), \quad (0,1) \ni \lambda = e^{-1/\tau}$$

The assertion that $\lambda \in (0, 1)$ is justified by our assumption that $\tau > 0$. So a training impulse $w(i)$ contributes to TE by its value, decreased by multiplying it by times some fixed number between 0 and 1 taken to the power of how many days it is from the current day. Doing this process for all of the training loads and summing, gets us a value that is proportional to the training effect. The k_1 and k_2 determine the scale of these effects and how the positive and negative effects are weighted against each other. This model asserts that all of these numbers are fixed.

From this, we can see some of the limitations of the model. First, it assumes that we recover the same way every day, which isn't true, since you can over train, not get enough sleep, get better at recovering over time, etc. Second, the modeling process is very specific. In practice, the model parameters depend on the person, the sport, and even the training regime. The parameters are not super interpretable; knowing just τ_1, for instance, does not give us any information about the athlete. The first problem can be solved by either expanding the model, adding more parameters to try to capture more the physiological process better. Second, we can allow the parameters to vary over time, which is what we are concerned with here.

The Time-Varying Fitness-Fatigue Model

We want to have recursive equations that allow the parameters to change over time. Breaking down either the fitness or the fatigue term in the model, we get that

$$\text{Fitness}(t) = k_1 \sum_{i=0}^{t-1} e^{t-i} w(i) = k_1 e^{-1/\tau_1} w(t) + k_1 e^{-2/\tau_1} w(t-1) + \cdots + k_1 e^{t/\tau_1} w(0)$$

Lets say that at time t_{new}, k_1 changes to k_{new} and τ_1 changes to τ_{new}. We would expect that the new model looks something like

$$\begin{aligned} \text{Fitness}(t) = {} & k_{new} e^{-1/\tau_{\text{new}}} w(t) + \cdots + k_{\text{new}} e^{-(t-t_{\text{new}})/\tau_{\text{new}}} w(t_{\text{new}} + \\ & + k_1 e^{-(t-t_{\text{new}})/\tau_{\text{new}} - 1/\tau_1} w(t_{\text{new}} - 1) + \cdots + k_1 e^{-(t-t_{\text{new}})/\tau_{\text{new}} - t_{\text{new}}/\tau_1} w(0) \end{aligned}$$

So, I propose a new model for performance:

$$p(t) = p_0 + \sum_{i=0}^{t-1} k_1^i \exp\left(- \sum_{j=i}^{t} (\tau_1^j)^{-1} \right) w(i) + \sum_{i=1}^{t-1} k_2^i \exp\left(- \sum_{j=i}^{t} (\tau_2^j)^{-1} \right)$$

where the superscript in the parameters indicates the parameter to be used for that day, i.e. k_1^i indicates the k_1 parameter on the ith day. This is better understood through the recursion equations given by $g(0) = h(0) = 0$, and

$$p(t) = p_0 + P(t) - N(t)$$

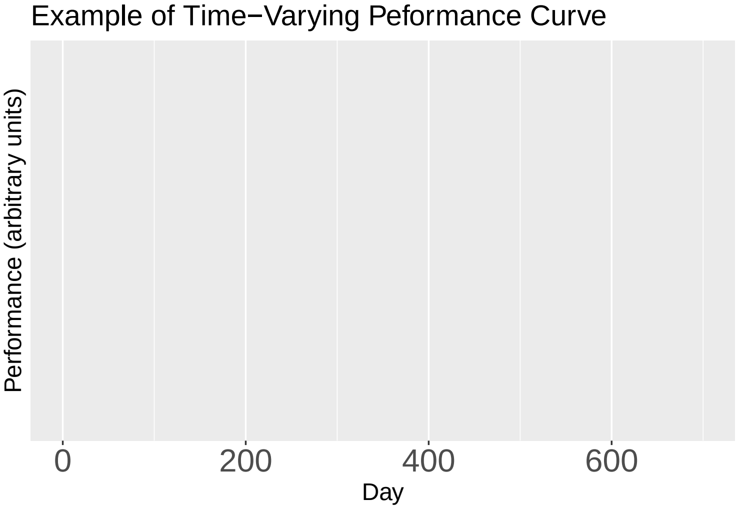
where

$$g(t) = e^{-1/\tau_1^t}(g(t-1) + k_1^t \cdot w(t)), \quad h(t) = e^{-1/\tau_2^t}(h(t-1) + k_2^t \cdot w(t))$$

These are the same recursion equations as the second ones in the previous section, except we introduced a dependency of the parameters across time. It is best to think of this model through the recursion equations. To get an idea as to what's going on, for values of parameters given by this table

	1-100	101-200	201-300	301-400	401-700
k_1	1	2	3	5	6
k_2	20	25	30	40	40
tau_1	2	4	6	8	10
tau_2	5	10	15	25	30

These parameters are not supposed to represent everying. Instead, they were chosen to get some change in the output, which is



It is important to note that these recursion equations capture that the performance is continuous despite the parameters changing wildly

Estimating the parameters

Let $z^{(t)} = (p_0^t, k_1^t, k_2^t, \tau_1^t, \tau_2^t)$ be the set of parameters on day t. For the sake of pseudo-code, set a cost function

$$C(\text{Error}(1, \ldots, i), w^{(t)}, w^{(t-1)}) = f(\text{Error}(1, \ldots, i)) + g(w^{(t)}, w^{(t-1)})$$

where $\text{Error}(1, \ldots, i)$ is the i dimensional vector that has the error for the 1 through ith performance days, f is some function that penalizes error, and the actual performance, like a scaler times SSE, MSE, or RMSE, and g is a function that penalizes the distance between the two parameter sets in some way. We wish to estimate the time varying parameters through Algorithm 1 .

Include here how exactly these algorithms work, this is still a work in progress.

```
Algorithm 1: Parameter estimation algorithm
  Data: Training Load, a vector w of length $n$
  Observed Performance, obs perf a vector of length $n$
  Initial Parameters, a vector $z^{(0)}$
  Settings for the Grid of values to search over, $S$
  Coeficent term for the Limit Cost term, a $\lambda > 0$
  Result: A vector of the predicted performance
  for $i = 1, \ldots, n$ do
      if Observed Performance at $i$ is a number then
          $G \leftarrow$ a grid of sets of parameters determined by some conditions
          $w^{(i)} = \text{argmin}_{w \in G} C(\text{Error}(1, \ldots, i-1), z, z^{(i-1)})$
          compute $P(i)$ with $w^{(i)}$
      else
          compute $P(i)$ with $w^{(i-1)}$
      end
  end
```

I propose here a choice of the function $g(\cdot, \cdot)$. This functions punishes the sets of parameters that have a different limit than the current set of parameters. To derive it we first restate the model

$$p(t) = p_0 + k_1 \sum_{i=0}^{t-1} e^{\frac{t-i}{\tau_1}} w(i) + k_2 \sum_{i=0}^{t-1} e^{\frac{t-i}{\tau_2}} w(i)$$

Assume that $w(i) = C$ for all i. This takes the convention that p_0 happens on day 0. Note that

$$\sum_{i=0}^{t-1} e^{\frac{t-i}{\tau_1}} = -1 + \sum_{i=0}^{s} \left(e^{-1/\tau_1}\right)^i$$

Finding the long-run limit of $p(t)$ then amounts to computing

$$\sum_{i=0}^{\infty} \left(e^{-1/\tau_1}\right)^i = \frac{1}{1 - e^{-1/\tau_1}}$$

Notice that this is a convergent geometric series, $e^{-1/\tau_1} < 1$ when $\tau_1 > 1$ (which we have assumed). Therefore,

$$\sum_{i=0}^{t-1} e^{\frac{t-i}{\tau_1}} = -1 + \frac{1}{1 - e^{-1/\tau_1}} = \frac{e^{-1/\tau_1}}{1 - e^{-1/\tau_1}}$$

Therefore

$$\lim_{t \to \infty} p(t) = p_0 + C \left(\frac{k_1 e^{-1/\tau_1}}{1 - e^{-1/\tau_1}} - \frac{k_2 e^{-1/\tau_1}}{1 - e^{-1/\tau_2}} \right)$$

So if $z = (p_0, k_1, k_2, \tau_1, \tau_2)$, and

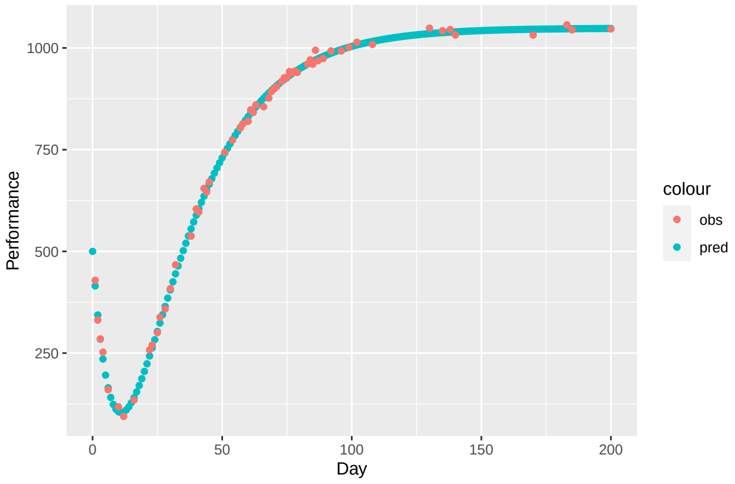
$$h(z) = \frac{k_1 e^{-1/\tau_1}}{1 - e^{-1/\tau_1}} - \frac{k_2 e^{-1/\tau_1}}{1 - e^{-1/\tau_2}}$$

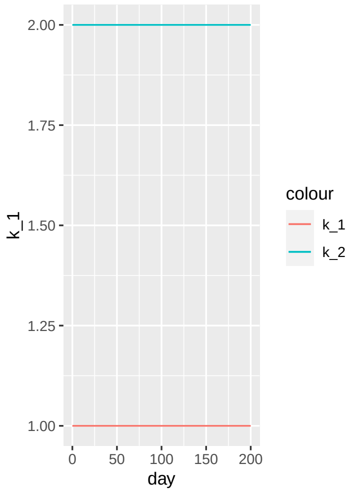
Then the proposed function g is

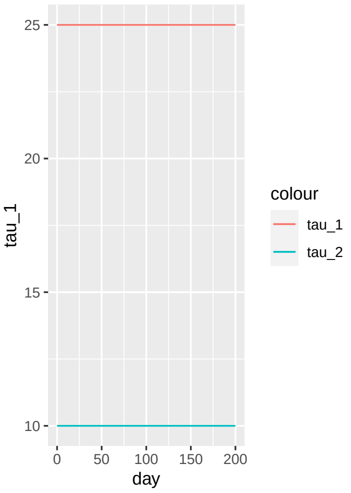
$$g(z, z') = |h(z) - h(z')|$$

Simulation Study

Will apply this algorithm to different simulated situations. I cannot put anything here because I am still working on improving the algorithm to it's final form. To make sure that it works. I will also do a comparative analysis to other methods to compare its effectiveness. I give here preliminary results of the algorithm when the function of error is root mean square error, and the function g is the one introduced before. The "actual performance" was simulated by first computing the time-invariant performance for the parameters, in the same order as the definition of z, $(500, 1, 2, 25, 10)$ and adding Gaussian noise to each point with mean 0 and standard deviation 10. The algorithm was run with initial parameters $(500, 1, 2, 25, 10)$ (the actual parameters), and $\lambda = .3$







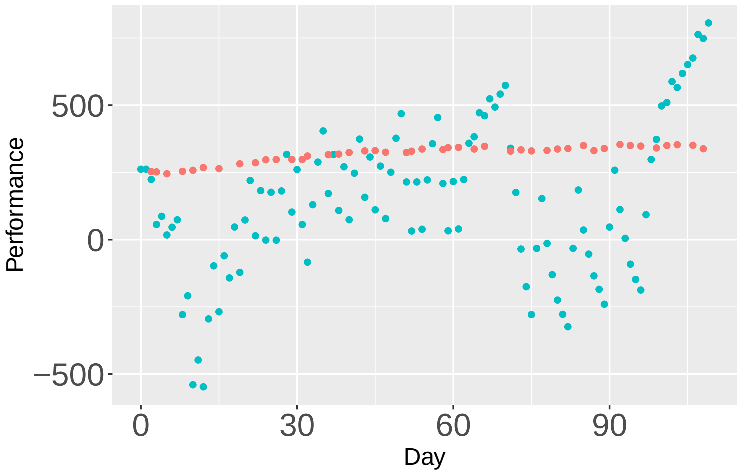
The plots show us that even though the algorithm predicted the correct parameters for all but one day, the modeled performance wandered off. There are some improvements to be made for the model.

Emperical Studies

I will do the same thing in the simulation study as to an empirical datasets, taken from Clarke and Skibba (2013). The algorithm was run with the initial parameters $(262, .1, .1, 11, 11)$.

```
## [1] 179.8288
```

```
## Warning: Removed 65 rows containing missing values (‘geom_point()‘).
```



What is happening here is that that the parameters that the algorithm searches over are all bad, so the algorithm doesn't change from it's original parameters. This results in a flat line. When the algorithm is improved upon, it will predict the performance better. I cannot, however, give anything better currently since the research has not been finished.

Conclusion

Hopefully it works!

Appendix

Derivation of the model

As in Clarke and Skibba, add later

Other items

There are probably going to be other items, like figures or datasets that I will not want to include in the paper directly.

Bibliography

Banister, Calvert, Savage, and Bach. 1975. "A Systems Model of Training for Athletic Performance." *Austr J Sports Med.*

Busso, Thierry, Robin Candau, and Jean-René Lacour. 1994. "Fatigue and Fitness Modelled from the Effects of Training on Performance." *European Journal of Applied Physiology and Occupational Physiology* 69 (January): 50–54. https://doi.org/10.1007/BF00867927.

Busso, Thierry, Christian Denis, Régis Bonnefoy, Andre Geyssant, and Jean-René Lacour. 1997. "Modeling of Adaptations to Physical Training by Using a Recursive Least Squares Algorithm." *Journal of Applied Physiology (Bethesda, Md. : 1985)* 82 (June): 1685–93. https://doi.org/10.1152/jappl.1997.82.5.1685.

Clarke, David C., and Philip F. Skiba. 2013. "Rationale and Resources for Teaching the Mathematical Modeling of Athletic Training and Performance." *Advances in Physiology Education* 37 (2): 134–52. https://doi.org/10.1152/advan.00078.2011.

Hellard, Philippe, Marta Avalos, Lucien Lacoste, Frederic Barale, Jean-Claude Chatard, and Gregoire P. Millet. 2006. "Assessing the Limitations of the Banister Model in Monitoring Training." *Journal of Sports Sciences* 24 (5): 509–20. https://doi.org/10.1080/02640410500244697.

Ludwig, Melanie, Alexander Asteroth, Christian Rasche, and Mark Pfeiffer. 2019. "Including the Past: Performance Modeling Using a Preload Concept by Means of the Fitness-Fatigue Model." *International Journal of Computer Science in Sport* 18 (1): 115–34. https://doi.org/doi:10.2478/ijcss-2019-0007.

Mujika, Iñigo, Jean-Claude Chatard, T. Busso, Andre Geyssant, F. Barale, and L. Lacoste. 1996. "Use of Swim-Training Profiles and Performance Data to Enhance Training Effectiveness." *Journal of Swimming Research* 11 (January): 23–29.

Thomas, Luc, Iñigo Mujika, and Thierry Busso. 2008. "A Model Study of Optimal Training Reduction During Pre-Event Taper in Elite Swimmers." *Journal of Sports Sciences* 26 (6): 643–52. https://doi.org/10.1080/02640410701716782.

Vermeire, Kobe, Michael Ghijs, Jan G. Bourgois, and Jan Boone. 2022. "The Fitness–Fatigue Model: What's in the Numbers?" *International Journal of Sports Physiology and Performance* 17 (5): 810–13. https://doi.org/10.1123/ijspp.2021-0494.

Winther, Andreas K. 2019. "Fitting Impulse Response Models in r." https://wintherperformance.netlify.app/post/banister-model/.