Predicting Cycling Performance from Historical Data

Anna Revinskaya
Stanford University
annare@stanford.edu

December 13, 2019

Abstract

We compare a Linear Regression and Neural Network approaches to estimate cycling times for casual cyclists on specific segments based on historical data. We further analyse which factors influence cycling speed the most.

1 Introduction

Cycling is a popular leisure and social activity for exercise or commute. Analyzing historical cycling data can help plan ETAs and set new exercise goals. In this paper, we explore factors that effect cycling performance on a specific road segment. To carry out this analysis, we look at weather conditions, ride frequency and attempt count. We then fit Linear Regression and Neural Network models to check if we can accurately predict future performance on unseen data.

2 Prior work

Most cycling performance studies are conducted based on professional cyclist data and often involve controlled experiments. Peiffer and Abbiss [12] used experimental data from cycling time trials completed by nine male cyclists in an environmental chamber and determined that performance is reduced under hot environmental conditions. Hilmkil et al. [10] trained LSTM based on time-series bike computer/sensor data collected during training sessions of professional cyclists to predict heart rate at any given time.

Casual cycling speeds have been studied in "Predicting Bicycle Travel Speeds Along Different Facilities Using GPS Data: A Proof-of-Concept Model." [8] which uses a least squares regression model to determine how type of cycling road (on-street, off-street, and mixed traffic) affects travel speeds.

	Page Mill (Moody to DF)	Baylands Park West
Avg grade	5.6%	0.0%
Distance (miles)	3.9	1.4
Approx. Direction	North-South	East-West

Table 1: Properties of the two segments we study: "Page Mill–Moody to drinking fountain" and "Baylands Park West" from strava.com website [3].

However, most casual cycling papers in the past have focused on the volume of cycling trips as opposed to speeds. For example, [13] determined that number of cycling trips increases relative to temperature and decreases relative to precipitation.

Our paper focuses on studying performance for casual cyclists on a specific road segment based on past attempts. We evaluate both a linear and a time series model. To model time series, we use an LSTM-based neural network [11].

3 Dataset and Features

3.1 Dataset

We combine weather and cycling data to generate our dataset.

We focus on two bike segments in the Bay Area: "Page Mill–Moody to drinking fountain" and "Baylands Park West" (see Table 1, Fig. 1, 2). We parse HTML pages on pagemilling.com [1] and ridewithgps.com [2] websites to collect Page Mill and

Figure 1: Map of "Page Mill–Moody to drinking fountain" from strava.com website [3]. This is a climbing segment in Santa Cruz Mountains.

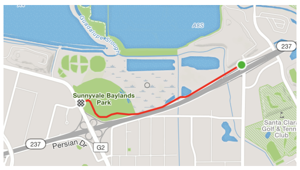


Figure 2: Map of "Baylands Park West" from strava.com website [3]. The segment stretches on along the South edge of San Francisco bay.

Baylands ride attempt information respectively.

Our weather data comes from "World Weather Online" website [5]. We download the data using wwo-hist [4] package for two zipcodes 94022 (for Page Mill) 94089 (for Baylands weather). The downloaded dataset contains temperature, humidity, wind speed and wind direction angle information at three hour increments and covers time span from 2009 to later part of 2019.

Cycling data is then matched with closest available weather measurements. Finally, we use a 80-20 split between train/cross-validation and test datasets.

We normalize both input feature and response variable data using Z-score:

$$X_{normalized} = \frac{X - \mu_X}{\sigma_X}$$

Note that we filter the data to discard attempts without heart rate data. We further discard attempt histories with fewer than 30 attempts when training per-cyclist models.

3.1.1 Dataset challenges

Working with cycling data presents multiple challenges (see Fig. 3):

1. Small dataset: just 30-40 data points for some cyclists.

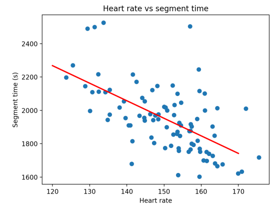


Figure 3: Segment completion time plotted against heart rate values for one of the cyclists. The red line corresponds to least squares fit to the data. As heart rate goes up, segment completion time goes down i.e. one gets faster.

Heart Rate	Heart rate measurement.
Attempt count	Number of attempts within the last 60 days before current attempt.
Temperature	Air temperature ($°C$).
Humidity	Humidity percentage.
Wind	Wind feature consists of two components: North and East. We compute projection of wind direction towards North and East using $cos(x)$ and $sin(x)$ functions respectively. We then multiply these component values by wind speed. Note that, "wind" values for South and West wind are represented as negative values.

Table 2: Input features.

2. Hidden information: no knowledge of other segment attempts, bike model, unforseen circumstances such as a flat tire, etc..

3. Noise: unreliable heart rate sensors, weather fluctuations between measurement stations and segment locations.

3.2 Input features

Table 2, includes the list of features as input to our models. Note that we also append a value of 1 to the feature vector to embed a bias term for our Ridge Regression model.

3.3 Output

Output of our model consists of a single value - time taken to complete the segment in seconds.

4 Method

4.1 Ridge Regression

Our first model is a Ridge Regression model. Since every person might have their own performance pattern, we are a training separate model for each cyclist. Loss function for cyclist j is defined as follows:

$$L^j = \frac{1}{|D^j_{train}|} \sum_{x^j, y^j \in D^j_{train}} (\theta^{jT} x^j - y^j)^2 + \lambda ||\theta^j||_2^2$$

where j is the user id, D^j_{train} is the training dataset for user j.
We then use the normal equation to compute θ^j.

$$\theta^j = (X^{jT} X^j + \lambda I)^{-1} X^{jT} y^j$$

4.2 Neural Network

Our second model is a Neural Network. Architecture of our network (see Fig. 4) is influenced by a few key observations:

1. Intuitively, performance is a time-series variable since we expect fitness level to gradually change over time. Therefore, the core of our architecture consists of LSTM units.

2. Raw cycling times are not the perfect predictor of future cycling times since each attempt is influenced by a number of factors (heart rate, weather, etc..). For e.g. biking at a slow pace while talking to a friend is not a good predictor of competing with a friend if we don't consider heart rate. Therefore, we first feed raw cycling times along with corresponding conditions to a dense layer in a way to "encode" them into data that consistently changes over time. Then, we "decode" the output of LSTM based on current conditions using another dense layer.

We use LeakyReLU activations for the Dense 1 and Dense 2 layers and ReLU for the LSTM units.

Since our data is noisy, we use Huber loss to reduce sensitivity to outliers:

$$L = \frac{1}{|D_{train}|} \sum_{x,y \in D} \begin{cases} \frac{1}{2} r^2, & \text{if } |r| < \delta \\ \delta |r| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \quad (1)$$

where $r = \theta^T x - y$.

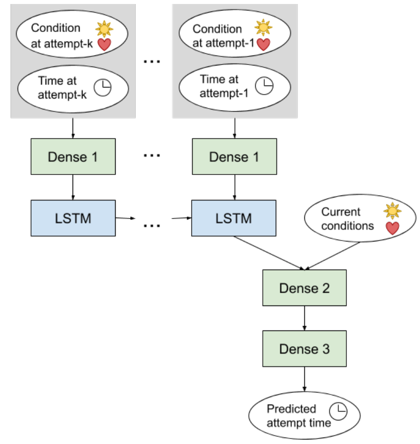


Figure 4: Neural Network architecture.

We pick $\delta = 1$.
Our model implementation uses Keras [7] with TensorFlow [6] backend.
We consider two model approaches.

4.2.1 Approach 1: Per-cyclist Neural Network model

This approach is similar to our Ridge Regression method where we also trained separate models for each cyclist. Equivalently, we average out errors we compute for each per-cyclist model.

4.2.2 Approach 1: Generalized model

Generalized model is trained against all cyclist data. Therefore, it must learn to differentiate cyclist performance purely based on the history we feed as an input.

5 Experiments

5.1 Parameter tuning

We use LOOCV to tune parameters for our linear model and 4-fold cross validation to tune parameters for our Neural Network.

Since our datasets can be very small, 4-fold cross validation produces widely-varied results when shuffling the dataset before splitting the folds. Therefore, we prefer LOOCV (Leave-one-out cross validation) for our Least Squares model where we can compute LOOCV error in one pass [9].

Unfortunately, we could not use LOOCV with our Neural Network since it is prohibitively time consuming. Instead we fall back to 4-fold cross-validation. You can see a loss function output for the first cross-validation fold in Fig. 5. We tune the regularization parameter to make sure validation and training loss stay similar to avoid overfitting.

Selected parameters are summarized below.

Ridge regression parameters
$\lambda = 0.6$

Per-cyclist Neural Network parameters
learning rate: 0.005, layer output sizes: (Dense 1: 3, LSTM steps: 2, Dense 2: 4), LSTM steps: 2, epochs: 100, regularization (Page Mill): 0.03, regularization (Baylands): 0.06.

Generalized Neural Network
learning rate: 0.007, layer output sizes: (Dense 1: 4, LSTM steps: 2, Dense 2: 6), regularization: 0.003, LSTM steps: 4, epochs: 100.

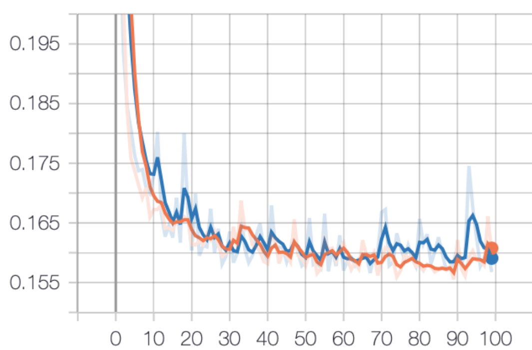
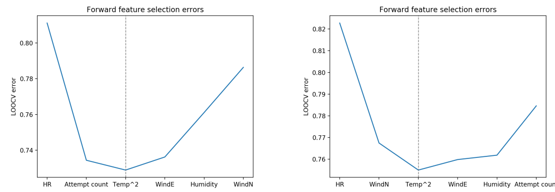


Figure 5: Train (orange) and validation (blue) for one of the cross-validation folds for generalized Page Mill segment model.

5.2 Feature selection

We ran feature selection for our Ridge Regression model (see Fig. 6) and determined that Heart Rate, Attempt Count and Temperature are the most important factors influencing times on Page Mill segment. On the other hand, most important factors on Baylands segment consist of Heart Rate, Wind North and Temperature. These results match our intuition since we expect to have wind coming from the bay and ocean to influence a flat segment (Baylands).



(a) Page Mill feature selection.(b) Baylands feature selection.

Figure 6: Forward feature selection results..

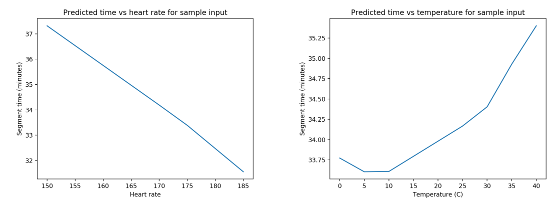


Figure 7: Predicted times plotted against heart rate (left) and temperature (right).

Note that we considered a linear feature for temperature as well as a squared feature. We find that cross-validation errors when using squared temperature are a little bit smaller.

Our Neural Network uses the same set of features we selected for the linear model with two exceptions: add "Wind East" feature to our Baylands network features and use temperature instead of temperature squared for both segments.

5.3 Model analysis

We run a few experiments to get an insight into the function learnt by our Generalized Neural Network. Specifically, we feed a sample input which is not a part of our dataset and vary one of the feature values. We find that the network finds a linear relationship with the heart rate input (Fig. 7). However, relationship learnt for temperature is non-linear. Specifically, the model finds that cycling time goes up when temperatures are too high and flattens out for low temperatures (Fig. 7).

6 Results

We compare performance of our models with a baseline. The baseline for per-cyclist models just

	Baseline	Ridge Reg.	NN
Test	0.0975	0.0824	0.0816
Train	0.1090	0.0826	0.0802

Table 3: Per-cyclist model errors for Page Mill segment.

	Baseline	Ridge Reg.	NN
Test	0.0762	0.0680	0.0676
Train	0.0689	0.0529	0.0555

Table 4: Per-cyclist model errors for Baylands segment.

	Baseline	Generalized NN
Test	0.1147	0.0928
Train	0.1097	0.0923

Table 5: Generalized model errors for Page Mill segment.

	Baseline	Generalized NN
Test	0.0726	0.0571
Train	0.0839	0.0622

Table 6: Generalized model errors for Baylands segment.

computes the mean segment completion time for the cyclist. Baseline model for Generalized Neural Network outputs the mean of the 4 historical values provided as input.

Results are compared based on normalized RMSE (Root Mean Squared Error).

$$NRMSE = \frac{\sqrt{MSE}}{\bar{y}}$$

where $\bar{y}$ is the mean of actual values

The error is averaged out when using multiple models (i.e. one for each cyclist).

Results for per-cyclist model comparisons are summarized in Tables 3 and 4. There are two things to note. First, Neural Network performance is comparable to our linear Ridge Regression model. This indicates that most relationships might be linear and time-series component is less relevant when training per-cyclist models. Second, some of the test errors are slightly smaller than training errors. This is probably due to small sizes of our test datasets that don't have enough "hard" examples.

Generalized Neural Network test data errors are presented in Tables 5 and 6. The generalized model outperforms baseline which indicates that it doesn't just average out historical values but instead learns a more complex function.

7 Conclusion / Future Work

Per-segment casual cycling data is extremely noisy and depends on many factors that we cannot observe. In spite of these challenges we successfully optimize models that outperform a mean-predicting baseline. At the same time we observe that our per-cyclist linear model has a comparable performance to the per-cyclist Neural Network. This indicates that given our small dataset we can only learn simple mostly-linear relationships. We also demonstrate that a general model trained across all cyclists does more than just summarize previous times. The model manages to learn that response variable has a linear relationship with heart rate and a non-linear relationship with temperature.
Acquiring a larger dataset would help train a more conclusive and general model.

Code link: `https://drive.google.com/file/d/1EPkWbcvoJxlCSxeoq7Bi3d6ss0LMKLD3/view?usp=sharing`

References

[1] Page mill counter. https://www.pagemilling.com/.

[2] Ride with gps. https://www.ridewithgps.com/.

[3] Strava application. https://www.strava.com/.

[4] wwo-hist pip package. https://pypi.org/project/wwo-hist/,.

[5] World weather online data api. https://www.worldweatheronline.com/developer/,.

[6] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro,

Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

[7] François Chollet et al. Keras. `https://keras.io`, 2015.

[8] Ahmed El-Geneidy, Kevin J. Krizek, and Michael J. Iacono. Predicting bicycle travel speeds along different facilities using gps data: A proof-of-concept model. 2007.

[9] Fleet Hertzmann. Cross validation. URL `http://www.cs.toronto.edu/~fleet/courses/C11/Handouts/CrossValidation.pdf`.

[10] Agrin Hilmkil, Oscar Ivarsson, Moa Johansson, Dan Kuylenstierna, and Teun Erp. Towards machine learning on data from professional cyclists. 08 2018.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.

[12] Jeremiah Peiffer and Chris Abbiss. Influence of environmental temperature on 40 km cycling time-trial performance. *International journal of sports physiology and performance*, 6:208–20, 06 2011. doi: 10.1123/ijspp.6.2.208.

[13] Sandar Tin Tin, Alistair Woodward, Elizabeth Robinson, and Shanthi Ameratunga. Temporal, seasonal and weather effects on cycle volume: an ecological study. *Environmental Health*, 11(1):12, Mar 2012. ISSN 1476-069X. doi: 10.1186/1476-069X-11-12. URL `https://doi.org/10.1186/1476-069X-11-12`.