Software tools for self-quantifier runners

Albert Ribas Admetlla

Master Thesis at HEIG-VD
Supervisor: Andrés Pérez-Uribe

Haute École d’Ingénierie et de Gestion du Canton de Vaud (HEIG-VD)
Institute for Information and Communication Technologies
Yverdon-les-Bains, Switzerland
July 2015
Abstract

The increasing availability of wearable sensors embedded in smartphones, watches and physical activity trackers has opened the door to original applications, mainly in health and wellness improvement.

The focus of this project is the analysis of the data collected by current commercially available devices providing real-time information about the speed and the heart rate frequency of the runner, and the elevation.

The goal is to develop a personalized data-driven model of the performance of a runner using his own historical training data. Such a model considers the performance of the runner during a race as a function of the current slope he/she is running and the accumulated effort since the start of the race. We suppose we know beforehand the profile of the race being realized. In particular, we found that a model that involves the median of the speed as a function of a given slope gives the best approach.
Acknowledgements

This thesis completes my education at the Haute École d’Ingénierie et de Gestion du Canton de Vaud (HEIG-VD), resulting in a Master of Science degree in Telecommunication Engineering. The research was carried out at the Information and Communication Technology Department, in HES-SO Yverdon-les-Bains, Switzerland.

I would like to express my gratitude to Professor Andrés Pérez-Uribe, my supervisor at the Department for offering to me this great opportunity and for hosting me in his research group. Also to Héctor Satizábal, colleague of the office, who has been actively helpful with all the issues around this thesis and whose advices have always encouraged me to go on.

I would also like to make a special emphasis to the rest of the people from the office, with whom I will not be able to forget the good moments while having the coffee breaks, while talking about random issues and playing to mini table tennis.

Additionally, I would like to thank all the people that I have met during my stance in Switzerland, with whom I have been able to share a lot of good moments, hoping not to be the last ones. Either the people from the apartment, with whom I have been living all these months while having some unforgettable moments, such as the “international dinners” and trips.

And finally, I would like to give a special thanks to my family, who from Barcelona have given me all the possible facilities and support to push forward this thesis. Needless to say that without you I would not have been able not to only finish this thesis, but to arrive to this moment. Thanks for all the efforts that during your life you have been doing and for all the good moments shared. For these reasons, and for others, I would like to dedicate this manuscript to you, hoping that you will enjoy it. Thanks for everything, and see you in a few.
Contents

Abstract i

Acknowledgements iii

1 Introduction 1
  1.1 Background ................................................................. 1
  1.2 Problem Statement .................................................... 1
  1.3 Thesis outline ............................................................ 2

2 State-of-the-art 3
  2.1 Self-tracking tools ...................................................... 3
  2.2 Pace analysis considering the elevation ............................. 4
     2.2.1 Heuristics ............................................................. 4
     2.2.2 GAP ................................................................. 5

3 Data 7
  3.1 Data format .............................................................. 7
  3.2 Data extraction .......................................................... 9

4 Data analysis 11
  4.1 Python in statistic analysis ........................................... 11
  4.2 Data without heart rate sensor ....................................... 11
  4.3 Data with heart rate sensor .......................................... 13

5 Case study: Race time estimation 17
  5.1 Case 1: Simple statistics ............................................... 18
  5.2 Case 2: Simple heuristics ............................................. 20
  5.3 Case 3: GAP ............................................................. 22
  5.4 Case 4: Speed and slope statistics .................................. 23
  5.5 Case 5: Speed and slope statistics on the moment of the race 25
  5.6 Case 6: Speed, slope and HRF statistics ............................ 28

6 Conclusions and further work 37
  6.1 Conclusions ............................................................... 37
  6.2 Further work ............................................................. 38

Bibliography 51
Chapter 1

Introduction

1.1 Background

One of the paradoxes of our time is that, while sedentary and poor eating habits ailments grow so does the concern about healthy habits. According to a study presented in April 2013 demonstrates the health benefits of physical activity [1]. From this idea borns the fact that in recent years, people of all ages has joined a growing fashion: running. Also motivated by the low cost of the material needed to develop this activity.

Companies have seen in this, a market opportunity that still was unexploited. Thus, stores, many road races in big cities, products of all kinds and even mobile applications and wearable devices have appeared.

It is not hard to see how more and more people join city races [13] and we can see more and more self-tracking devices to monitor distance, time and speed such as watches or smartphones. Moreover, easy access to these devices has made the amateur sector been professionalized giving rise to more accurate devices with new features and ways to improve the runner performance.

Thus, self-tracking in sport is getting a significant boom, due to, not only the possibility to calculate the amount of physical exercise, but also its quality. Most of applications allow runners to set objectives of, for example, distance covered or calories burned per week. It is what is called “gamification” [15].

1.2 Problem Statement

The main shortcoming of today’s applications is the lack of customization offered. Most of them only show the instantaneous value of basic parameters like speed, pace, distance...Without actually give a tool to predict these parameters in the future.

Most of the runners are always looking to improve the performance of each race regarding previous races. If we can give them a tool to know beforehand what time do they need to finish a particular race given the profile of that race, the runners can plan a better strategy particularized for each race and therefore train in specific slope conditions. Also we can provide the remaining time in any moment of the race. That is useful to adapt, in real time, the strategy of the race.

That is why this manuscript seeks to provide a new tool to give personalized feedback to the runner to improve their performance in races by highlighting the profile of each race.
1.3 Thesis outline

Chapter 2 features the state-of-the-art, focusing on the applications that are being used nowadays and their lack that this project is trying to fill. Chapters 3 and 4 describe the data that we have worked with and the preliminary analyses. Chapter 5 describes and presents the results of the different estimation models to estimate and predict the time taken by a runner to finish a race. Finally Chapter 6 presents the final conclusions of the study.
Chapter 2

State-of-the-art

As discussed in the previous chapter, the market of mobile applications has seen a considerable increase in the catalog of applications for sport and wellness. And, in recent years, the apps that have the task to improve your sporting experience have proliferated to the point that poised to surpass activity trackers revenue [10].

In this Chapter we describe the state of the art for the existing tools that we can find in the market for self-tracking in the context of sport activities. Furthermore we describe the pace analysis in which we base some of the models described in Chapter 5.

2.1 Self-tracking tools

There are two big groups in which we can classify the trackers that can be found on the market: hardware devices and mobile applications. Both are equipped with basic functionalities to calculate: time, distance, pace and speed.

Within the group of hardware devices there are, on the one hand, devices that can be found in form of watches with GPS receiver as for example brands like Garmin\(^1\) and Polar\(^2\). They both have a broad catalogue of products for outdoor sports including running and cycling. These solutions were the first to appear in the market and, although they are more expensive than mobile apps, it is possible to see in city races the majority of the runners wearing these watches. Moreover, the costumer has the option to acquire a heart rate sensor as a complement that gives an added value to the watch.

On the other hand, there are a wide variety of sensors that communicate with the smartphone via an specific mobile app. The principal characteristics are the tracking of the time, distance, speed and calories burned of the runner. A clear example of these sensors are: Nike+ Sensor\(^3\) and Adidas miCoach\(^4\).

Within the group of mobile applications, we can find a broad variety of sports and wellness apps and now they are on the top of most downloaded apps [2]. The most popular apps for running tracking are: Endomondo\(^5\) is one of the best known. It can automatically detect when you stop during the race and is even able to notify hydration needs. Runtastic, like the previous one, is able to record pace, distance, calories, altitude and training time and sharing on Twitter and Facebook. Nike+, for its part, is which has a more attractive design. It very suited to beginners because the level of each runner and challenges encourage users to overcome challenges. Nike+

---

\(^1\)http://www.garmin.com
\(^2\)http://www.polar.com
\(^3\)http://www.nikeplus.com.br
\(^4\)http://micoach.adidas.com
\(^5\)https://www.endomondo.com
pioneered these applications and still has a large number of users. With Sports Tracker⁶ you can track and monitor progress, store the data on a daily basis and see the races on the website. Meanwhile, RunKeeper⁷ allows you to add photos, see progress, add objectives, or receive alerts. As above, this has a social character as it allows sharing and watching the activities, achievements and plans of our friends. Strava⁸ (Android, iOS) adds some features to the usual run-tracking recipe to make it competitive and gamified. The app records your running speed, distance traveled, time and course taken, but also combine it with leaderboards, achievements, and challenges. Strava supports a variety of running trackers, in addition to Android Wear.

2.2 Pace analysis considering the elevation

There have been many studies on human behavior when facing changes in the slope of the land. One of the first studies on the physiological effects of running downhill and was made by Davies [4] based on the work of Margaria et al. [7], and he stated that “the relationship between the net oxygen intake and speed of running uphill and downhill was essentially linear and thus for both forms of exercise are, for a given gradient, the aerobic cost of running per unit distance covered is constant and independent of speed”. That is, the increase of the speed requires more oxygen.

Based on these studies have been proposed different heuristics to determinate the equivalent pace on a flat land.

2.2.1 Heuristics

Naismith’s rule

William W. Naismith, a Scottish mountaineer, proposed in 1892 a rule of numb to help in the planning of a walking expedition by calculating how long it will take to walk the route, including ascents [17]. This rule was named as Naismith’s rule and is as follows:

Allow 1 hour for every 5 kilometres (3.1 mi) forward, plus 1 hour for every 600 metres (2000 ft) of ascent.

Figure 2.1: A plot of walking speed versus slope resulting from Naismith’s rule and Langmuir corrections [6] for base speeds of 5 km/h and 4 km/h compared to Tobler’s hiking function. “Hiking speed” by Darekk2 - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons.

⁶http://www.sports-tracker.com
⁷http://runkeeper.com
⁸https://www.strava.com
The basic rule assumes hikers of reasonable fitness, on typical terrain, under normal conditions. Alternatively, the rule can be used to determine the equivalent flat distance of a route. This is achieved by recognising that Naismith’s rule implies an equivalence between distance and climb in time terms.

Figure 2.1 shows the evolution of walking speed as a function of the slope following the Naismith’s rule.

**Mervyn Davies**

British researcher Mervyn Davies (1980/81) conducted treadmill tests which showed that for each 1% of upgrade, elite runners slowed by about 3.3%.

Applying the same percentage to runners of all speeds, this translates as follows: 10 seconds per mile for 5-minute milers, 15 seconds per mile for 7 minute-milers and 20 seconds per mile for 10-minute milers.

Looking at this from an elevation change perspective, a 1 percent grade climbs 52.8 feet per mile. Each 100 feet of climbing is equivalent to one mile of 2 percent grade – whether it comes in a single steep pitch or spread out over several miles. In other words, each 100 feet of climbing costs the best athletes a little less than 20 seconds. Marginal Boston qualifiers lose about 30 seconds, and 10-minute milers lose about 40 seconds.

Running downhill, Davies also tested downgrades and found that descents don’t give back as much as ascents take away. In other words, that rate at which you make up time going down hill is less than the rate of time you loose going uphill. The average difference is about 45%. This means that running downhill allows you to make up only about 55% of the time you lost going uphill.

### 2.2.2 GAP

Grade Adjusted Pace (GAP) is a Strava’s tool that estimates the equivalent flat land pace for a given running pace on hilly terrain based on the research of C.T.M. Davies [4, 3]. Running uphill requires more effort than running on a flat grade, so GAP adjusts pace to be faster than the actual running pace. Similarly, GAP is slower than actual pace on downhill terrain.

When the user uploads his training sessions to Strava, GAP is automatically calculated for every 1 km of distance.
Chapter 3

Data

The database we used in our analysis comprises a total of 55 races of the same runner with distances ranging from approximately 8 to 20 km, with variable slopes.

We have divided this database into two separate groups to study different estimates: training with heart rate receiver and training without heart rate receiver.

To record the data, we used the following commercial devices: Garmin Forerunner 610 watch for the study without the heart rate sensor and a watch Garmin Forerunner 620 with heart rate receiver for the study of heart rate. Both devices allow the measurement of time, speed, distance, pace, elevation and GPS positioning.

3.1 Data format

The format in which the device collects data is based on XML files with a tree structure by tags. There are two types of files: '.tcx', whith all the variables used in the study, and "+.gpx", with the coordinates of the position by GPS.

GPX File

GPS Exchange Format (GPX), is an XML schema designed as a common GPS data format for software applications. It can be used to describe waypoints, tracks, and routes. The format is open and can be used without the need to pay license fees. Location data (and optionally elevation, time, and other information) are stored in tags and can be interchanged between GPS devices and software. Common software applications for the data include viewing tracks projected onto various map sources, annotating maps, and geotagging photographs [16].

TCX File

Training Center XML (TCX) is a data exchange format introduced in 2007 as part of Garmin’s Training Center product. The XML is similar to GPX since it exchanges GPS tracks, but treats a track as an Activity rather than simply a series of GPS points. TCX provides standards for transferring heart rate, running cadence, bicycle cadence, calories in the detailed track [18]. It also provides summary data in the form of laps. Code 3.1 shows a segment of a TCX file as example.

<?xml version="1.0" encoding="UTF-8"?>
<TrainingCenterDatabase xsi:schemaLocation="\http://www.garmin.com/xmlschemas/TrainingCenterDatabase/v2\http://www.garmin.com/xmlschemas/TrainingCenterDatabasev2.xsd"
CHAPTER 3. DATA

<xlmns:ns5="http://www.garmin.com/xmlschemas/ActivityGoals/v1"
xlmns:ns3="http://www.garmin.com/xmlschemas/ActivityExtension/v2"
xlmns:ns2="http://www.garmin.com/xmlschemas/UserProfile/v2"
xlmns="http://www.garmin.com/xmlschemas/TrainingCenterDatabase/v2"
<Activities>
   <Activity Sport="Running">
      <Id>2015-01-18T10:19:08.000Z</Id>
      <Lap StartTime="2015-01-18T10:19:08.000Z">
         <TotalTimeSeconds>345.206</TotalTimeSeconds>
         <DistanceMeters>1000.0</DistanceMeters>
         <MaximumSpeed>3.3959999084472656</MaximumSpeed>
         <Calories>67</Calories>
         <AverageHeartRateBpm>
            <Value>145</Value>
         </AverageHeartRateBpm>
         <MaximumHeartRateBpm>
            <Value>160</Value>
         </MaximumHeartRateBpm>
         <Intensity>Active</Intensity>
         <TriggerMethod>Manual</TriggerMethod>
         <Track>
            <Trackpoint>
               <Time>2015-01-18T10:19:08.000Z</Time>
               <Position>
                  <LatitudeDegrees>46.54207086190581</LatitudeDegrees>
                  <LongitudeDegrees>6.669112639501691</LongitudeDegrees>
               </Position>
               <AltitudeMeters>716.4000244140625</AltitudeMeters>
               <DistanceMeters>1.5199999809265137</DistanceMeters>
               <HeartRateBpm>
                  <Value>98</Value>
               </HeartRateBpm>
               <Extensions>
                  <TPX xmlns="...">
                     <Speed>0.0</Speed>
                     <RunCadence>89</RunCadence>
                  </TPX>
               </Extensions>
            </Trackpoint>
         </Track>
      </Lap>
   </Activity>
<Author xsi:type="Application_t">
   <Name>Garmin Connect API</Name>
   <Build>
      <Version>
         <VersionMajor>15</VersionMajor>
      </Version>
   </Build>
</Author>
</Activities>
3.2. DATA EXTRACTION

To extract the data from these files, we have implemented a python script that reads the values of the labels from TCX files and writes them in a text file in a tabular form with extension “.tab” for further processing and analysis.

As can be seen on Table 3.2, we only store the values for time, elevation, speed and heart rate frequency.

To manage the data in Python we use a kind of data structure from the Pandas library called Dataframe. It is a 2-dimensional labeled data structure comprised of rows and columns.

In order to calculate the slope, we use a Simple Linear Regression model. This model finds the relationship between a dependent variable (elevation) and an explanatory (distance) within a sliding window of 6 samples, approximately equivalent to 100 m (Figure 3.1).

### Table 3.1: Description of subtags within the parent tag `<Lap>`.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;TotalTimeSeconds&gt;</code></td>
<td>Lap time in seconds</td>
</tr>
<tr>
<td><code>&lt;DistanceMeters&gt;</code></td>
<td>Distance covered on a lap (default 1000 m)</td>
</tr>
<tr>
<td><code>&lt;MaximumSpeed&gt;</code></td>
<td>Maximum speed on a lap in m/s</td>
</tr>
<tr>
<td><code>&lt;Calories&gt;</code></td>
<td>Calories burned on a lap</td>
</tr>
<tr>
<td><code>&lt;AverageHeartRateBpm&gt;</code></td>
<td>Average heart rate on a lap in bpm</td>
</tr>
<tr>
<td><code>&lt;MaximumHeartRateBpm&gt;</code></td>
<td>Maximum heart rate on a lap</td>
</tr>
<tr>
<td><code>&lt;Track&gt;</code></td>
<td>Parent label where we find each of the samples of each lap and in which are recorded each the instantaneous values of time, elevation, distance, speed and heart rate.</td>
</tr>
</tbody>
</table>

3.2 Data extraction

TCX files always start with the `<Activities>` tag. The kind of sport is specified by `<Activity Sport="">`. The `<Id>` tag saves the starting date and time of the activity, while the `<Lap StartTime="">` defines the date and time of the start of each lap. The system can be configured to record data for a fixed interval distance. This interval is set by default to 1000m.

The main sub tags within `<Lap>` are described in Table 3.1.
Table 3.2: Example of a tab-file content. The number of samples depends on the duration of the activity. This example consists on 561 samples for a one-hour activity.

Figure 3.1: Example of Linear Regression for the Distance-Elevation for a random training session. In (–) it is the actual profile and in (--) it is the predicted model.
Chapter 4

Data analysis

This Chapter presents the development of the data analysis. It has been divided into two sections: data analysis without the heart rate sensor and data analysis with the heart rate sensor.

All scripts we have used have been written in the programming language Python.

4.1 Python in statistic analysis

Python is a widely used general-purpose, high-level programming language[5, 11, 12]. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code, then it would be possible in languages such as C++ or Java [14, 8] The language provides constructs intended to enable clear programs on both a small and large scale.

We decided to use this tool because of its wide use in many fields, particularly in the analysis of data and statistics. The great flexibility offered and the huge number of existing libraries make it a very appropriate tool for our study. Also it has an open-source license called Python Software Foundation License 1.

Thanks to NumPy, SciPy and Matplotlib libraries we can use Python for scientific computing. It allows us to make use of related vectors and matrices much like the style of Matlab operations.2

4.2 Data without heart rate sensor

In this first study of the data without the heart rate sensor we started with a dataset of 35 races in which we have chosen 33 for training and 2 for testing.

The goal of the project is to give an estimation of the time the runner will need to finish the race. To do so, we process the training dataset by creating bins of speed but, instead of counting the frequency like an histogram, we calculate the percentiles of the data within each bin.

Later, we use the test dataset and calculate a table of what we have denominated the effort of the runner as follows:

1. For each slope-speed pair from the test set, we find the corresponding bin in the histogram of training slopes.

2. For that bin, we can extract the index of the percentil associated to the actual speed. We interpretate this index as the measure of the effort. In the case where there is no corresponding bin, we either take the first bin if the value of the test slope is smaller than any slope in the training set, or the last bin if the value is actually larger.

1https://docs.python.org/2/license.html
2https://www.h2desk.com/blog/python-scientific-computing/
3. To make the prediction of the estimated time to finish the race, we take the value of each percentile for the corresponding bin. Thereby we find the speed that the runner should have for each percentile.

4. Take the distance between two consecutive samples of the race \((\Delta d_{i-1,i})\), and divide by predicted speed for each percentil \((v'_{i})\), to obtain the estimated time between samples \((\Delta t'_{i-1,i})\):

\[
\Delta t'_{i-1,i} = \frac{\Delta d_{i-1,i}}{v'_{i}}
\]

5. Do the summation for each column (percentile) we find the time taken to finish the race according to the desired percentile.

Table 4.1 shows an example of the results that can be obtained using the aforementioned algorithm.

As it can be seen, Figure 4.1 shows the estimated time for the test set called “Bern”. As the effort increases, the speed increases and therefore the time needed to end the race decreases.

For example, for the test of Bern, assuming we set an effort of 60%, we obtain the following results:
4.3. DATA WITH HEART RATE SENSOR

You want to finish in 85.7 minutes
The application will set the effort to 60 %

Figure 4.2 shows the evolution of the effort, the estimated time given a pre-fixed effort and the real and predicted time throughout the race. We can appreciate when the runner’s speed is below or above the predicted speed.

![Graph showing the evolution of effort, estimated time, real and predicted time throughout the race.]

Figure 4.2: From top to bottom: evolution of the effort, the estimated time given a pre-fixed effort and the real and predicted time throughout the race.

4.3 Data with heart rate sensor

To analyze the data with the heart rate frequency (HRF), we take the records collected by a Garmin Forerunner 620 as mentioned before and constitute a total of 16 races spread between Train and Test.

We did different tests to observe the behavior of the heart rate frequency as a function of the other variables we already knew (speed, slope) in order to find some correlation.

The first test was to observe boxplots of the evolution of the HRF and speed as a function of the slope. Races have been divided into two parts to see if there was a remarkable difference of
variability depending on the moment of the race. Furthermore, we have removed the first 20%, which is when the runner is adapting. As the reader can observe in the Figure 4.3, during the first half of the race there is no clear trend. But as it is seen, as slope increases, also increases the HRF. This behavior is clearer in the second half where the cumulative effort of the runner is present.

![Figure 4.3: Boxplot HRF and speed as a function of slope. In green there is the HRF and in blue there is the speed.](image)

Another test was plotting heatmaps of the heart rate frequency as a function of the speed and the slope for all the the dataset dividing into 4 windows: 0%-25%, 25%-50%, 50%-75% and 75%-100% of the total distance. Using all races gave unreliable results. Then, we only used races between 11 to 12 km to have lower variability.

In Figure 4.4 we can observe the high variability of the heart rate at the beginning of the race. It doesn’t give us an idea of the variation in heart rate while we increase or decrease speed or while we increase or decrease the slope.

For that reason we did another test, also dividing the whole dataset in four parts, to plot the variation of HRF (ΔHRF) every 100 m. We discarded the first quarter of the datasets to focus on the part of the race that is less sensible to big changes. As it can be seen in Figure 4.5, for lower speeds and negative slopes, the HRF decreases, while on the other hand, when slope and speed increase, the HRF also increases.
Figure 4.4: Heatmap of HRF as a function of the speed and the slope for the filtered dataset for (a) first quarter of the race, (b) second quarter of the race (c) third quarter of the race and (d) last quarter of the race.

Figure 4.5: Heatmap of $\Delta$HR as a function of the speed and the slope for the filtered dataset. From left to right, (a) second quarter, (b) third quarter and (c) last quarter of the race.
Chapter 5
Case study: Race time estimation

The present chapter presents an evaluation of the different models of the performance of the runner as a function of several parameters such as slope, accumulated effort and heart rate frequency and compare their capability of estimating the race time of a series of validation races given a set of training sessions.

We have evaluated a total of 6 models, that in where the models 1 to 5 do not consider the use of the heart rate frequency whereas the last model takes into account this parameter in order to evaluate if the fatigue has a weight in terms of time prediction.

As said in the previous chapter, we started with a dataset of training sessions with distances that range from 10 to 13 km and subsequently they are compared to all training sessions recorded with heart rate frequency sensor.

To compute the error of the estimation we have calculated the difference between the estimated time and the real time. The estimated time for a given sample $i$ can be obtained as

$$t'_i = t_{i-1} + \Delta t'_{i-1,i}$$

Where $\Delta t'_{i-1,i}$ is the elapsed time between the actual and the previous sample calculated from the elapsed distance and the estimated speed and $t_{i-1}$ is the elapsed real time until the last sample.

We can also compute the Remaining Time from sample $i$ as

$$RT_i = \sum_{k=i}^{N} \Delta t'_{k-1,k}$$

The test dataset used to validate the estimator is formed by three races labeled as: Feeling Bad, Morat-Fribourg and Test. The first one is a training session in which the runner had a high variation on the heart rate. The second test is the most hilly and ends in higher elevation than in the beginning. The last test example, is a random test with a normal behaviour compared to all training dataset. See Figure 5.1 to notice the profile of each training session from the test dataset. Morat-Fribourg ends in a higher elevation than it the start, while the variation of the elevation for Felling Bad and Test is nearly zero.

In the Appendix we compile all the plots for the estimation error as a function of the distance of the race and compared to the elevation and speed for each race of the test set in every case studied.
CHAPTER 5. CASE STUDY: RACE TIME ESTIMATION

Figure 5.1: Training session profile for (a) Feeling bad, (b) Morat-Fribourg and (c) Test. Horizontal axis represents the distance in kilometres and in vertical axis is represented the elevation in metres.

5.1 Case 1: Race time estimation using a simple statistics

In this first case we only use the median speed of the historical records of training sessions. To obtain the predicted time we divide the difference of the distance between samples by the median speed and then compute the summation of all samples in order to get the estimated time at the end of the race. See Table 5.1 for comparison of real time and predicted time for each validation race.

<table>
<thead>
<tr>
<th>Case</th>
<th>Distance (km)</th>
<th>Race time (Hr:Min:Sec)</th>
<th>Predicted time (Hr:Min:Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>0:52:36</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>1:30:17</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>0:59:58</td>
</tr>
</tbody>
</table>

Table 5.1: Distance, race time and predicted race time for each race of the dataset in Case 1.

As it can be seen, Figure 5.3 shows that the predicted time is the same as the race time. And Figure 5.4 shows that the median of error of the estimation for the test dataset is zero. Therefore, the estimator performs well on all the scenarios.

We can say that the model based on the median speed of the historical records is a good estimator when the runner is trained and the races of the test set are similar to the training set. At the end, uphill and downhill slopes are compensated and that makes that the median speed doesn’t be affected.

If we add more races to the training dataset, the median speed of these races tends to stabilize around a certain value as can be seen in Figure 5.2.

It would take to us to do more tests in the same conditions but with untrained runners to see if we obtain the same trend.
5.1. **CASE 1: SIMPLE STATISTICS**

(a) Median speed for the filtered training dataset (tends to 3.17 m/s).

(b) Median speed for the unfiltered training dataset (tends to 3.21 m/s).

Figure 5.2: Evolution of the median speed as a function of the number of races within the training set.

Figure 5.3: Race time (–) and Predicted time (--) measured in minutes using the median speed of the training datasets for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 1. Horizontal axis represents the distance in km.

Figure 5.4: Error between race time and estimated time for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 1. The value on the top of each boxplot is the median.
5.2 Case 2: Race time estimation using simple heuristics

To determine a pace over each of these mile segments, we’re going to use some rules developed by Dr. Mervyn Davies, referenced in Tim Noakes’ book ‘Lore of Running’ [9]. We’re going to use six rules of thumb for calculating the pace you need for each mile (Table 5.2).

<table>
<thead>
<tr>
<th>Distance (mi)</th>
<th>1-16</th>
<th>16-21</th>
<th>21-26.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every 1 % of upgrade</td>
<td>pace + 3.3 %</td>
<td>pace + 3.8 %</td>
<td>pace + 4.3 %</td>
</tr>
<tr>
<td>Every 1 % of downgrade</td>
<td>pace - 1.8 %</td>
<td>pace - 1.8 %</td>
<td>pace - 1.7 %</td>
</tr>
</tbody>
</table>

Table 5.2: Rules for pace adjustment proposed by Noakes for every 1 % of upgrade or downgrade of the slope as a function of the distance of the race in miles.

Since training dataset is always between 10 and 13 km, we will only use the first conversion. To adjust the speed we have looked for what would be the flat land pace by converting the current pace as a function of the uphill or downhill slope. To do so, we have computed the correction factor for each slope as

For \( \text{slope} > 0 \):

\[
C = \frac{1}{1 + 0.033s}\]

For \( \text{slope} < 0 \):

\[
C = \frac{1}{1 + 0.018s}\]

As the correction factor is referred to the pace, to calculate the predicted elapsed time, we divide the elapsed distance by the product of the historical median speed and the inverse of the correction factor.

\[
\Delta t_{i-1,i} = \frac{\Delta d_{i-1,i}}{v'C}
\]

That gives us the prediction time before starting the race. See Table 5.3 for comparison of real time and predicted time for each validation race.

<table>
<thead>
<tr>
<th>Case 2</th>
<th>Distance (km)</th>
<th>Race time (Hr:Min:Sec)</th>
<th>Predicted time (Hr:Min:Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>0:52:13</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>1:27:39</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>0:58:53</td>
</tr>
</tbody>
</table>

Table 5.3: Distance, race time and predicted race time for each race of the dataset in Case 2.

Figure 5.5 shows that the predicted time is practically the same as the race time like in the previous case and in Figure 5.6, the reader can see that the median of the error is also very close to 0.
Figure 5.5: Race time (–) and Predicted time (--) measured in minutes using the flat land equivalent pace for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 2. Horizontal axis represents the distance in km.

Figure 5.6: Error between race time and estimated time for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 2. The value on the top of each boxplot is the median.
5.3 Case 3: Race time estimation using Grade Adjusted Pace

We use a model that predicts and estimates the finishing time by using an approach similar to the Grade Adjusted Pace (GAP) used by Strava.

We don’t know the algorithms that use Strava so, we build a model based on a set of training sessions downloaded from Strava’s website in order to reproduce the same adjustment that they use.

We start with a set of eight training sessions that include the pace, the pace adjusted and the variation of the elevation for every 1 km of distance.

From this set we fit a linear model with the slope (calculated as the variation of the elevation divided by 1 km of distance) and the ratio GAP/Pace. As the reader can see in Figure 5.7 the linear model fits the data with a correlation value of 0.95.

![Figure 5.7: Linear regression for GAP/Pace ratio](image)

The formula used to get the estimated GAP is the following:

\[ \text{GAP'} = \text{Pace} \cdot (-0.04 \cdot \text{Slope} + 0.98) \]

Where the values of the slope and the intercept were get from the linear model.

Similarly to the estimation of the previous section, to compute the predicted elapsed time we first have to find the correction factor given by the slope and the intercept of the linear model. Then, we divide the elapsed distance by the product of the inverted factor and the historical median speed to find the predicted race time. See Table 5.4 for comparison of real time and predicted time for each validation race.

<table>
<thead>
<tr>
<th>Case 3</th>
<th>Distance (km)</th>
<th>Race time (Hr:Min:Sec)</th>
<th>Predicted time (Hr:Min:Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>0:51:40</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>1:28:43</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>0:58:56</td>
</tr>
</tbody>
</table>

Table 5.4: Distance, race time and predicted race time for each race of the dataset using the model of Case 3

Again, predicted values are quite close to the real ones and as can be seen in Figure 5.8, the curve of the predicted time follows the real time.
5.4. CASE 4: SPEED AND SLOPE STATISTICS

Figure 5.9 shows the error of the estimated time for the test set. It performs better than Case 2 in terms of the median.

![Figure 5.9](image)

Figure 5.8: Race time (–) and Predicted time (--) measured in minutes using the flat land equivalent pace for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 3. Horizontal axis represents the distance in km.

![Figure 5.8](image)

Figure 5.9: Error between race time and estimated time for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 3. The value on the top of each boxplot is the median.

![Figure 5.9](image)

5.4 Case 4: Race time estimation using speed statistics and slope

We use our first model that indicates the median speed given a slope. This model considers that that speed can be achieved independently of the accumulated effort, whether during the first part of the race, after 3 km, 8 km or 12 kms of running.

This model maps the historical speed in slope intervals and then computes the median of speed for each interval.

To compute the elapsed predicted time we proceed as explained in Chapter 4. For each slope-speed pair from the test set, we find the corresponding bin in the histogram of training slopes. We take the associated median speed and then we divide the elapsed distance between two consecutive slopes by each predicted speed and compute the summation for all the samples.
As can be seen in Table 5.5 this model performs similarly in all scenarios. It is, the distribution of the error is the same for all the test sets.

<table>
<thead>
<tr>
<th>Case 4</th>
<th>Distance (km)</th>
<th>Race time (Hr:Min:Sec)</th>
<th>Predicted time (Hr:Min:Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>0:52:51</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>1:30:36</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>0:59:59</td>
</tr>
</tbody>
</table>

Table 5.5: Distance, race time and predicted race time for each race of the dataset using the model of Case 4.

Figure 5.10 shows the comparison between the race time and predicted time for each test set. As the reader can appreciate, both curves are practically overlapped. Thus, the errors shown in Figure 5.11 are approximately 0.

Moreover, method used in Case 4 gives the best approach to the predicted running time we’ve seen so far. Compared to Case 1, Case 4 is more accurate. Thus, using the historical median speed given a slope is a more precise method - in terms of predicted running time - than only using the historical median speed, as we are adding more information.

Figure 5.10: Race time (-) and Predicted time (--) measured in minutes using the flat land equivalent pace for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 4. Horizontal axis represents the distance in km.
5.5. **CASE 5: SPEED AND SLOPE STATISTICS ON THE MOMENT OF THE RACE**

We use a model that provides the expected speed of the runner during the next 100m as a function of the slope and the part of race: 0 - 25%, 25 - 50%, 50 - 75% or 75 - 100%

With this model we analyse the local variabilities in order to understand better the behaviour of the runner in each part.

To compute the estimated elapsed time we treat each part of the race independently as if it was a small race. Furthermore, we proceed as in Case 4, estimating the median speed for each part. Then the distance between consecutive slopes is divided by the predicted speed. See Table 5.6 for the results of the prediction in each distance-window for each race of the test set.

Split the race in four quarters permits us to identify in which quarter the runner is running over or below the median. Therefore we can identify if the runner started too fast and the consequent fatigue in the future. As can be seen in Figure 5.12, Morat-Fribourg is a clear example of that. The runner started fast, but in the course of the race he finishes later than what we predicted. Feeling Bad shows that in the middle-half of the race slows his speed respect to the estimated speed due to physical problems. Then, the runner maintains a constant speed until the end of the race. Test is the most regular of the three studied races. The runner is almost always running at the expected median speed. Therefore the curves of race time and predicted time are overlapped.

Figure 5.13 shows the distribution of the error throughout the race for each race of the test set. We can see an homogeneous distribution for all races of the test set except in the second quarter of Morat-Fribourg where the error is remarkably different.
### Table 5.6: Distance, race time and predicted race time for each race of the dataset using the model of Case 5.

<table>
<thead>
<tr>
<th>Case</th>
<th>Distance (km)</th>
<th>Race time (Hr:Min:Sec)</th>
<th>Predicted time (Hr:Min:Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeling bad</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 - 25 %</td>
<td>2.48</td>
<td>0:15:07</td>
<td>0:14:04</td>
</tr>
<tr>
<td>25 - 50 %</td>
<td>2.49</td>
<td>0:14:59</td>
<td>0:14:12</td>
</tr>
<tr>
<td>50 - 75 %</td>
<td>2.48</td>
<td>0:12:20</td>
<td>0:12:47</td>
</tr>
<tr>
<td>75 - 100 %</td>
<td>2.49</td>
<td>0:11:59</td>
<td>0:12:14</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 - 25 %</td>
<td>4.29</td>
<td>0:23:35</td>
<td>0:24:28</td>
</tr>
<tr>
<td>25 - 50 %</td>
<td>4.27</td>
<td>0:22:07</td>
<td>0:24:11</td>
</tr>
<tr>
<td>50 - 75 %</td>
<td>4.27</td>
<td>0:23:16</td>
<td>0:21:54</td>
</tr>
<tr>
<td>75 - 100 %</td>
<td>4.29</td>
<td>0:22:55</td>
<td>0:21:17</td>
</tr>
<tr>
<td>Test</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 - 25 %</td>
<td>2.84</td>
<td>0:16:01</td>
<td>0:16:03</td>
</tr>
<tr>
<td>25 - 50 %</td>
<td>2.84</td>
<td>0:15:33</td>
<td>0:16:06</td>
</tr>
<tr>
<td>50 - 75 %</td>
<td>2.82</td>
<td>0:14:22</td>
<td>0:14:32</td>
</tr>
<tr>
<td>75 - 100 %</td>
<td>2.85</td>
<td>0:13:44</td>
<td>0:14:03</td>
</tr>
</tbody>
</table>

Figure 5.12: Race time (–) and Predicted time (--) measured in minutes. From top to bottom first row (a) corresponds to Feeling Bad, second row (b) corresponds to Morat-Fribourg and last row (c) corresponds to Test for the model used in Case 5. In columns, from left to right there are the corresponding windows from 0 to 100 % of the race distance. Horizontal axis represents the distance in km.
Figure 5.13: Error between race time and estimated time for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for the model used in Case 5. Indices 1-4 indicate the quarter of the race. The value on the top of each boxplot is the median.
5.6 Case 6: Race estimation as a function of speed, slope and heart rate frequency

We use two different models that provide the expected speed of the runner as a function of the slope and the current HR frequency. Model 1, similar to the model used in case 4, maps the HRF into a matrix of slope and speed intervals as a 2-D histogram, but we also compute the mean and the standard deviation of HR frequencies instead of just counting the values. On the other hand, Model 2 uses a built-in function of matplotlib library called griddata that fits a surface of HR frequency given the speeds and slopes of the training dataset and then interpolates this surface with the intervals and slopes that we have set.

As can be seen on Figure 5.14, when we have a small population of training sets, Method 1 is not able to fill the matrix for all values of slope and speed. However, Method 2 (Figure 5.16) performs better since it interpolates the missing values instead of assigning a NaN (not a number)\(^1\).

Figure 5.14: Heatmap of mapped HR frequency using Method 1 on the filtered training dataset. From left to right: mean, standard deviation and count for the filtered dataset. In the lower axis, the intervals (bins) of the slope and in the left axis the intervals (bins) of speed.

In order to get more data in our mapping matrix, we decided to use all the training sets (i.e. not filtering by distance). As can be seen on Figure 5.15, the mapping matrix has more values and that helps to predict better the HRF as explained next. On the other hand Method 2 does not improve significantly.

In order to estimate the HRF, we first created an estimator for each model to predict the HRF given a certain slope and speed. For each pair of slope and speed sample, we search for the corresponding value of HRF in the predictions matrix. Then we compute the error between the actual HRF and the predicted.

Table 5.7 shows that griddata-based method performs better than our implementation. As the matrix of Method 1 has several missing values, we decided not to filter the datasets by the distance of the training sessions in order to have a bigger database and fill the prediction matrix. In Table 5.8 we can see that the number of samples that Method 2 can predict has remained approximately still while for our estimator is nearly doubled. Even so, we have not improved regarding to Method 2.

Figure 5.18 shows that the error for the found HRF in Method 1 is significantly higher than Method 2. By considering the non-filtered dataset, we achieve error reduction in Method 2 but not in Method 1 (Figure 5.19).

\(^1\)https://en.wikipedia.org/wiki/NaN
5.6. CASE 6: SPEED, SLOPE AND HRF STATISTICS

Figure 5.15: Heatmap of mapped HRF using Method 1 on the entire training dataset. From left to right: mean, standard deviation and count for the non-filtered dataset. In the lower axis, the intervals (bins) of the slope and in the left axis the intervals (bins) of speed.

Figure 5.16: Heatmap of HRF using `griddata` function on the filtered training dataset.

Figure 5.17: Heatmap of HRF using `griddata` function on the non-filtered training dataset.

Therefore, we can say that interpolation is a better estimation since with few training sessions gives better results in terms of error.

Now to predict the race time we proceed in a similar way to heart rate prediction and Case 4. We iterate over each slope sample of the test race, we search for the interval that belongs in our prediction matrix and then for that row (slopes) we compute the minimum of the heart rates with the actual heart rate to find the index of the columns (speed) and this gives us the interval of the speed. We use the mean of that interval to obtain the predicted speed for that heart rate frequency.

Once we have all the predicted speeds, we calculate the elapsed predicted time in the same way as all previous cases and therefore the total predicted time (Table 5.9 - 5.10).

The prediction of the total race time does not improve as we increase the number of runs of the database but it does from the Method 1 to Method 2. As can be seen on Figure 5.20-5.21, the model we used in this case gives the worst prediction of the time for all the scenarios and both methods used.

As can be seen, Figure 5.22-5.23 show the distribution of the estimated time error for each
CHAPTER 5. CASE STUDY: RACE TIME ESTIMATION

<table>
<thead>
<tr>
<th>Method 1</th>
<th>Method 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found 222</td>
<td>Found 448</td>
</tr>
<tr>
<td>Total 506</td>
<td>Total 506</td>
</tr>
<tr>
<td>Found 321</td>
<td>Found 798</td>
</tr>
<tr>
<td>Total 811</td>
<td>Total 811</td>
</tr>
<tr>
<td>Found 234</td>
<td>Found 556</td>
</tr>
<tr>
<td>Total 562</td>
<td>Total 562</td>
</tr>
</tbody>
</table>

Table 5.7: Number of samples that each method is able to predict in the test races for the filtered dataset.

<table>
<thead>
<tr>
<th>Method 1</th>
<th>Method 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found 361</td>
<td>Found 453</td>
</tr>
<tr>
<td>Total 506</td>
<td>Total 506</td>
</tr>
<tr>
<td>Found 635</td>
<td>Found 800</td>
</tr>
<tr>
<td>Total 811</td>
<td>Total 811</td>
</tr>
<tr>
<td>Found 416</td>
<td>Found 556</td>
</tr>
<tr>
<td>Total 562</td>
<td>Total 562</td>
</tr>
</tbody>
</table>

Table 5.8: Number of samples that each method is able to predict in the test races for all training dataset.

Method 1 and Method 2 perform better than using all training sets. On the other hand, Method 2 remains with lower errors in both cases.
Figure 5.18: Error between actual HRF and predicted for (a) Feeling bad, (b) Morat-Fribourg and (c) Test on the filtered training dataset for the model used in Case 6. Indices 1-2 indicate the method used. The value on the top of each boxplot is the median.

**Table 5.9:** Distance, race time and predicted race time for each race of the dataset using Method 1 and Method 2 of Case 6 on the filtered training dataset.

<table>
<thead>
<tr>
<th>Case 6 (Filtered)</th>
<th>Distance (km)</th>
<th>Race time (Hr:Min:Sec)</th>
<th>Predicted time (Hr:Min:Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>1:37:33</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>2:37:30</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>1:48:42</td>
</tr>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>1:17:12</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>1:54:07</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>1:39:58</td>
</tr>
</tbody>
</table>

**Table 5.10:** Distance, race time and predicted race time for each race of the dataset using Method 1 and Method 2 of Case 6 on the not-filtered training dataset.

<table>
<thead>
<tr>
<th>Case 6 (Unfiltered)</th>
<th>Distance (km)</th>
<th>Race time (Hr:Min:Sec)</th>
<th>Predicted time (Hr:Min:Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>1:37:04</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>2:44:50</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>1:46:13</td>
</tr>
<tr>
<td>Feeling bad</td>
<td>10.01</td>
<td>0:54:43</td>
<td>1:24:50</td>
</tr>
<tr>
<td>Morat-Fribourg</td>
<td>17.18</td>
<td>1:32:14</td>
<td>2:24:50</td>
</tr>
<tr>
<td>Test</td>
<td>11.41</td>
<td>1:00:01</td>
<td>2:08:31</td>
</tr>
</tbody>
</table>
Figure 5.19: Error between actual HRF and predicted for (a) Feeling bad, (b) Morat-Fribourg and (c) Test on the non-filtered training dataset for the model used in Case 6. Indices 1-2 indicate the method used. The value on the top of each boxplot is the median.
Figure 5.20: Race time (–) and Predicted time (--) measured in minutes using the flat land equivalent pace for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for Method 1 and 2 used in Case 6 on the filtered dataset. Horizontal axis represents the distance in km.
Figure 5.21: Race time (–) and Predicted time (--) measured in minutes using the flat land equivalent pace for (a) Feeling bad, (b) Morat-Fribourg and (c) Test for Method 1 and 2 used in Case 6 on the non-filtered dataset. Horizontal axis represents the distance in km.
Figure 5.22: Error between race time and estimated time for (a) Feeling bad, (b) Morat-Fribourg and (c) Test on the filtered training dataset for the model used in Case 6. Indices 1-2 indicate the method used. The value on the top of each boxplot is the median.

Figure 5.23: Error between race time and estimated time for (a) Feeling bad, (b) Morat-Fribourg and (c) Test on the non-filtered training dataset for the model used in Case 6. Indices 1-2 indicate the method used. The value on the top of each boxplot is the median.
Chapter 6

Conclusions and further work

6.1 Conclusions

As explained in the introduction of this thesis, it has been carried out an exhaustive study to develop a personalized data-driven model of the performance of a runner using his own historical training data. Thanks to it, we have been able to get a better understanding of the behavior of a running during the race.

Specifically, we have paid special attention to changes on slope and on heart rate frequency, which have an important impact on the performance. Indeed, most of the estimators we used were built using these variables.

Furthermore, the estimators designed have been focused on heart rate and speed prediction, features that are notably important if we plan to achieve the goal we set, being able to predict the time that a runner needs to finish a race.

For our first experiments, we reported on test using several training sessions of the analysed runner without taking into account the heart rate frequency. Those experiments have indicated that the estimator that considers the median speed given a certain slope, gives the best estimation and the lowest error. Therefore the use of a model that adds more information (slope) reduces the error and is more accurate.

On the other hand, for the experiments which take into account the heart rate frequency, we have developed several predictors using more complex estimations of the speed and those report worse estimations and the highest error. One of the main reasons is that we don’t have enough variability of slopes and speeds in our training dataset. Thus we would need to obtain more races for the training set in different conditions.

Moreover, increasing the number of training sessions for our dataset does not improve the prediction of the time in terms of error respect to the test dataset. While considering heart rate frequency and slope we perceive an important improvement, since we can fill the prediction matrix with more samples.

Another important observation is that the first model used in Case 1, median speed of the entire training dataset, reported similar results as Case 4. The difference in race time prediction is less than 20 seconds between both models. For example, for Morat-Fribourg, the race time was 1:32:14 and the predicted race time was 1:30:17 for Case 1 and 1:30:36 for Case 4 respectively. That is 2,17% and 1,82% of error respectively.

A possible explanation that the model based on the median speed gives such good results is that the runner analysed was trained. Thus the median speed in most of the races was the same.
6.2 Further work

While this thesis has demonstrated that we can predict the finishing race time beforehand, many opportunities for extending the scope of this thesis remain. This section presents some of these directions:

1. Considering that we have worked with the training sessions of a trained runner. It would be of great interest to study whether if we can also predict the evolution of the performance of an un-trained runner. This means that if our predictors perform well in any kind of scenario.

2. Improve HRF estimation algorithm in order to have a better representation of the effort of the runner along the race.

3. Optimize the code so it can run faster than the current one and better optimization of the memory.

4. Adapt the Python code in order to be used for some mobile applications.
Appendix

This Appendix contains the plots of the error between the real time and the estimated time along the race.

We have represented for each studied model, the profile of the race, the speed, the median of the speed of the training dataset and the error.
Figure 1: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Case Study 1.
Figure 2: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Case study 2.
Figure 3: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Case study 3.
Figure 4: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Case study 4.
Figure 5: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Feeling Bad Case study 5.
6.2. FURTHER WORK

Figure 6: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Morat-Fribourg Case study 5.
(a) Test first quarter of the race.

(b) Test second quarter of the race.

(c) Test third quarter of the race.

(d) Test last quarter of the race.

Figure 7: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Test Case study 5.
6.2 FURTHER WORK

<table>
<thead>
<tr>
<th>Distance (km)</th>
<th>Median Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>3.17</td>
</tr>
<tr>
<td>650</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td></td>
</tr>
<tr>
<td>750</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td></td>
</tr>
<tr>
<td>850</td>
<td></td>
</tr>
<tr>
<td>900</td>
<td></td>
</tr>
<tr>
<td>950</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Elevation [m]</th>
<th>Speed [m/s]</th>
<th>Time Error [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.3</td>
<td>1.25</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>2.0</td>
<td>1.7</td>
<td>1.75</td>
</tr>
<tr>
<td>2.5</td>
<td>1.9</td>
<td>2.0</td>
</tr>
<tr>
<td>3.0</td>
<td>2.1</td>
<td>2.25</td>
</tr>
<tr>
<td>3.5</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td>4.0</td>
<td>2.5</td>
<td>3.0</td>
</tr>
<tr>
<td>4.5</td>
<td>2.7</td>
<td>3.5</td>
</tr>
<tr>
<td>5.0</td>
<td>2.9</td>
<td>4.0</td>
</tr>
</tbody>
</table>

(a) Feeling Bad Method 1 filtered runs.

(b) Feeling Bad Method 2 filtered runs.

(c) Feeling Bad Method 1 non-filtered runs.

(d) Feeling Bad Method 2 non-filtered runs.

Figure 8: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Feeling bad Case study 6.
Figure 9: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Morat-Fribourg Case study 6.
Figure 10: Elevation, speed and error of the estimated time along the distance of the race for each test in dataset. Test Case study 6.
Bibliography


