



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

*Distributed
Computing*



Fitness 2.0 - Modeling Fitness and Heart Rate

Master Thesis

Dominic Langenegger

`dominicl@student.ethz.ch`

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:

Pascal Bissig

Klaus-Tycho Förster

Prof. Dr. Roger Wattenhofer

May 13, 2015

Abstract

There exist many smart phone applications allowing people to track their exercise sessions. However, most of them only show generic information like speed or estimated calories burned. Our thesis addresses heart rate evolution during exercise and proposes two heart rate models that can accurately predict the heart rate during a workout by only observing the values of speed, cadence, and grade. The superior one includes a novel application domain for a linear-chain Conditional Random Field and allows us to predict the heart rate with a mean relative error of 5 % during a running session. The models are trained on individually collected workout data, as well as publicly available exercise data from the Endomondo online fitness platform. A trained model allows us to better understand the influences on the heart rate of subjects and their fitness level. Based solely on the trained model parameters, we also introduce a novel way to tell subjects when their fitness level increases.

Contents

Abstract	ii
1 Introduction	1
2 Related Work	3
2.1 External Influences on Energy Consumption	3
2.2 Selecting Music for Workout Sessions	5
2.3 Heart Rate Models	6
2.4 Fitness Models	7
2.5 BeatRunner	8
3 Methodology	11
3.1 Influences on Heart Rate	11
3.2 Heart Rate Model	13
3.3 Fitness Measure	14
3.4 Evaluation	15
4 Data Sets	16
4.1 Individual Collection	16
4.2 Online Data	27
5 Heart Rate Model	35
5.1 Data Pre-processing	35
5.2 Model Approaches	39
5.3 Fitness	46
6 Results	51
6.1 Heart Rate	51
6.2 Fitness	66

7 Discussion	68
7.1 Data Sets	68
7.2 Heart Rate Model	69
7.3 Future Work	70
Bibliography	71
A Theory Details	A-1
A.1 Boxplots	A-1
B More Experiment Results	B-1
B.1 Hardware	B-1
B.2 Conditional Random Field (CRF)	B-1
C Code Notes	C-1
C.1 Google Elevation API	C-1
C.2 Step Detection Algorithm	C-2
D Data Set Details	D-1
D.1 Endomondo	D-1
E Glossary	E-1
E.1 Acronyms	E-1
E.2 Glossary	E-1

Introduction

Running is probably the most common form of exercise. Many people around the globe go for a run multiple times a week to keep their body fit and healthy. With the fast growth of the smart phone market, many mobile applications were introduced to help users track their exercise sessions. Some of them remind users to regularly work out, some provide detailed information about workout aspects, like a map with the running route or speed measurements over time. But most of this information is generic and does not reflect the underlying effects of exercise.

We use workout data collected with off-the-shelf smart phones to understand more about the evolution of the heart rate and fitness during exercise. Based on observation features like speed, cadence and grade, a heart rate model is proposed. As we aim to understand the coherence between observation value changes and the heart rate during exercise, we deliberately do not include the heart rate itself as part of the model input parameters to predict future changes.

This thesis is influenced by previous work on the BeatRunner fitness tracking application by Valerius Huonder and Monika Graf. [2, 3] The application aims at selecting the right music during workout sessions, in order to control the heart rate of a user. A major challenge of this task is to correctly predict future changes of the heart rate, to which our proposed heart rate models, give a possible solution.

Our main contributions are:

- A novel use of a linear-chain Conditional Random Field model to accurately predict the heart rate during running sessions, based solely on speed, cadence and grade measurements.
- A fitness improvement function, to tell subjects when their fitness level increases, based solely on trained heart rate model parameters.
- A large collection of workout data from multiple disciplines, extracted from the publicly available workouts on the Endomondo online fitness platform.

We discuss related work, with a focus on external influences on the energy consumption during workouts, and fitness models, in Chapter 2. In Chapter 3 we introduce the methodology used to build our heart rate models, and deduct a fitness measure. An extensive discussion of the collection process for our workout data sets follows in Chapter 4. Chapter 5 introduces the core concepts of the proposed models, which are then evaluated in Chapter 6. The final discussion in Chapter 7 concludes this thesis.

Related Work

In this chapter, we discuss related and previous work. It is split in five parts, starting with mostly medical literature about energy consumption during exercise. In Section 2.2, we discuss some work about automatically selecting the music during workout sessions. Section 2.3 and 2.4, discuss heart rate and fitness models, respectively.

With the BeatRunner Android app, there exists previous work by students of the Distributed Computing Group at ETH Zurich, aiming at the perfect music selection during workouts. We present the app in Section 2.5.

2.1 External Influences on Energy Consumption

While many factors play a role in changes of energy consumption during a workout session, we are especially interested in body-external (i.e. not concerning body internal chemical processes) components that are measurable using off-the-shelf smart phones. A major influence is the speed, which is often investigated in coherence with at least one other influence and is therefore part of many studies. In this section, we motivate the importance of the energy consumption based on its close relationship to the heart rate in aerobic exercise and discuss literature about some main factors of energy consumption.

The energy consumption of humans (and animals) is closely related to their oxygen (O_2) consumption. The heart pumps oxygen rich blood to the body cells which is used in cellular respiration, the process of converting energy from nutrients into adenosine triphosphate (ATP). ATP is the main energy source for most processes in a human body and is also used for muscular activity and therefore body movements. This process was first described by Keilin in 1925 [4] and Lohmann in 1929 [5]. It is observed in common aerobic exercise and shows why the heart rate (and therefore the amount of oxygen-rich blood pumped through the circulatory system) is directly comparable to the energy consumption in the human body. [6] This relation motivates the use of the heart rate as an energy measure but also shows why medical tests often measure the oxygen

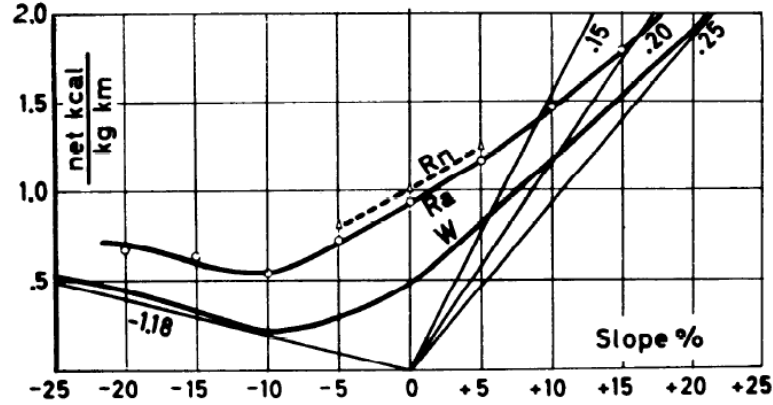


Figure 2.1: Energy expenditure per kg of body mass and km as a function of grade for walking (W) and running in athletes (R_a) and non-athletes (R_n). Also shown are isoefficiency lines as a measure of the mechanical energy efficiency. Figure courtesy of [8].

consumption of a subject to allow for more precise statements about their overall energy consumption.

2.1.1 Grade

When Margaria studied the energy cost of walking and running in 1938 [7], he showed that the cost of running is a linear function of speed (as opposed to a curve when walking). Later in [8] from 1963, he closely investigated the impact of grade running on energy consumption for grades from -10% to 15% . He confirmed that even with varying grades, the energy expenditure is a linear function of speed and showed how untrained subjects need around 5% to 7% more energy than trained athletes. The analysis of energy expenditure in running as a function of incline showed that less energy is needed as the grade gets smaller until a value of -10% , where the cost of running starts increasing again slightly.

In 1974, Davies et al. confirmed the observation that the aerobic cost of running per distance is constant and independent of speed. [9] In their paper *The Physiological Responses to Running Downhill*, they investigated the metabolic cost of a single male subject running gradients from -45% to 5% at varying speeds from 6 km h^{-1} to 18 km h^{-1} .

Alberto et al. investigated the energy cost of running at extreme uphill and downhill slopes of up to $\pm 45\%$ in 2002. [10] The paper confirms the findings of Margaria and shows an extended version of the energy cost as a function of the grade. We use these findings in Section 5.1.4 to derive an intensity factor (Figure 5.3) from the energy measurements for running.

A similar study on five human subjects was also published by Minetti (a

co-author of [10]) et al. in 1994 [11]. It focused on finding the most economical gradient for running and concluded it to be about -10% .

2.1.2 Wind

Especially with increasing speed, we expect wind to become a significant factor in the energy consumption of a running subject. Davies studied the effect of wind assistance and resistance while running at different speeds in a wind tunnel in 1980. [12] He confirmed earlier findings of Pugh [13, 14] that the energy cost of running with head wind is proportional to the square of the wind speed. Davies showed, that there exists a close relationship between the energy expenditure of increased head wind speed and increased grade, concluding, that horizontal (wind) and vertical (slopes) forces have the same impact on the energy cost of running, both independent from the speed.

2.1.3 Step Frequency

The impact of the step frequency on the energy consumption during running has not been studied as much as grade impact, but some reference work exists. In 1988 and 1990, Cavagna et al. [15, 16], studied the connection between the step frequency and the step-average power output during running. Using data from 170 runs by five subjects, he showed, that the freely chosen step frequency while running does only slightly increase with increased speed. For example the average step frequency increased by 20% when doubling the running speed (from 10 km h^{-1} to 20 km h^{-1}), but was virtually constant for speeds below 10 km h^{-1} .

We conclude, that the energy consumption while running is well covered in literature, especially for speed and grade. The important insights are, that the cost of running is a linear function of speed and the grade follows a higher order relation. In our setting, the measurement of wind speed and direction is very hard, especially since it is very location specific and highly dependent on factors like distraction and shielding.

2.2 Selecting Music for Workout Sessions

Multiple systems to control workout sessions via auditory feedback have been developed and described in the literature. They all rely on the thesis, that a runner adapts his step frequency to the beats per minute (bpm) of the audio signal played, as suggested in [17].

In 2006, Nuria from Microsoft Research and Flores-Mangas introduced *MPTrain* [18], a mobile phone based system that allows the user to specify a desired exercise pattern (heart rate over time). It assists the user by (i) constantly

monitoring physiology and movement; and (ii) selecting and playing suiting music to encourage the user to change his speed or keep it, in order to stay on track with the selected exercise pattern.

MPTrain tracks the heart rate and step frequency to select a previously unplayed song to be played next. 10 s before a song ends, the next one is selected to have its bpm value as close as possible to the desired step frequency of the user. The desired step frequency is selected to have a similar relative difference to the actual step frequency, as the one between actual heart rate and desired heart rate. For those calculations, the heart rate is projected into a percentage of the interval between the user's resting and maximum heart rate. (Both values have to be specified manually by the user)

Two years later, Nuria et al. presented *TripleBeat* [19], a follow-up paper of *MPTrain*, which directly compares the new system to the old one. The main differences are: (i) The system proposes a workout schedule based on a user selected high level goal (e.g., burn more fat); (ii) a visual feedback to help the user stay in the proposed heart rate training zone; (iii) a virtual competition with other runners; and (iv) a novel score function to quantify the user's performance. The used heart rate training zones were previously described by [20] and were supposedly introduced by Haskell and Fox in 1970 [21].

While both systems by Nuria et al. show interesting parallels to the BeatRunner application (see Section 2.5), they do not try to understand the underlying reasons for heart rate changes in detail and have a very simplistic method of choosing new songs based on a direct tie of bpm to the step frequency.

With *PaceGuard* by Fortmann et al. from 2012, there exists another mobile phone application to support runners in managing their workout by auditory feedback. [22] The focus of this work is to help beginners and inexperienced runners to keep their cadence constant during a workout via rhythmic pulse beats. The appropriate bpm value is determined by finding the step frequency based on accelerometer data during the first 150 s of a run.

2.3 Heart Rate Models

Under the term heart rate model, we understand a mathematical model with some observation input, that produces an estimate of the heart rate of a subject as an output.

In 2009, Koenig et al. presented a model for heart rate prediction during lokomat walking. [23] For the medical rehabilitation process in which a Lokomat is typically used, the speed of the treadmill is capped at 3.2 km h^{-1} . As an input to the model, the treadmill speed and power exchanged between the patient and the Lokomat are used. The proposed model, can foresee the temporal evolution of the heart rate and is used to ensure, the heart rate stays in an optimal range.

Koenig proposes to use the first few minutes of a Lokomat session to train the model parameters on the patient, enabling the model to take full control of the rest of the session by accurately predicting heart rate changes.

A heart rate model based on a evolutionary neural network, was proposed by Xiao et al. in 2010 [24] and refined in 2011 [25]. The main goal of this work was to predict the heart rate at some time $t + \Delta t$, given the heart rate and 3D accelerometer data from time t during normal daily activity (not only workouts). We note, that in contrary to many other model approaches, the current heart rate is also used for the prediction step. The absolute error in bpm of heart rate was about 5 ± 8 for a 30 s prediction during a 90 minute session of natural daily activity.

2.4 Fitness Models

In their 1990 paper, Shvartz et al. [26], performed an extensive literature review of studies where the maximal oxygen (O_2) uptake $VO_2 \max$ was measured. As described in [27], the cardiac output can be formulated as a fraction of absorbed oxygen per time and the difference of oxygen levels in arterial (oxygen rich) and venous (oxygen poor) blood — the so-called arteriovenous O_2 difference — as seen in Equation (2.1).

$$\text{Cardiac output} = \frac{O_2 \text{ absorbed per minute by the lungs } [\text{ml min}^{-1}]}{\text{Arteriovenous } O_2 \text{ difference } [\text{ml L}^{-1}]} \quad (2.1)$$

Therefore, the maximum oxygen uptake $VO_2 \max$ is a good indicator of the aerobic physical fitness of an individual. This value has been shown to be dependent on the body mass of the subject and decreases linearly between age 9 and 75. [26] It is commonly measured relatively to body mass in $\text{ml kg}^{-1} \text{min}^{-1}$ or absolutely in L min^{-1} . Many physicians and physiologists proposed methods to determine $VO_2 \max$ and the resulting cardiac output. Fick [28] first proposed an exact method by measuring the oxygen intake using a spirometer and the oxygen concentration in blood taken from the pulmonary vein and intravenous cannula¹, during maximal effort of a subject.

Estimates to calculate $VO_2 \max$ include the Cooper test [29], which uses the distance in meters a subject can run in 12 min to approximate $VO_2 \max$. Other approaches use a multi-stage fitness test [30] or just values for maximum HR_{\max} and resting heart rate HR_{rest} as in [31] by Niels et al. in 2003. The evaluation showed that Niels et al. were able to estimate $VO_2 \max$ with a mean error of 4.5 %, when experimentally determining HR_{\max} and HR_{rest} .

¹Both blood vessels connect the heart to the lung

Although $VO_2 \max$ has been shown to be a good measure of physical fitness, it is hard to determine its value only from data during exercise sessions. Even if using a method like proposed in [31], both HR_{\max} and HR_{rest} are hard to determine since both values are not commonly reached during an exercise session. The resting heart rate is usually measured in the morning, before getting up from bed and the maximum heart rate has to be reached in a training session at maximum intensity. A popular estimate for HR_{\max} is based on the formula $HR_{\max} = 220 - \text{age}$, which has however shown to be rather inaccurate - mean errors of 7 – 11 bpm have been shown. [32, 21, 33]

Based on the literature review by Shvartz et al. [26], Polar Electro, a large manufacturer of electronic sports training equipment, developed their *Running Index*. [34, 35] It consists of a single integer in the range from 0 to 100, where higher values signify higher fitness. Based on a subjective assessment by a regular user of a Polar fitness tracker², the index is fluctuating heavily and prone to overemphasize physically challenging workouts compared to regular balanced ones.

2.5 BeatRunner

The BeatRunner Android application is a project maintained by the Distributed Computing Group at the Computer Engineering and Networks Laboratory of ETH Zurich. It was developed by Valerius Huonder [2], based on previous work by Monika Graf [3]. The application aims to improve endurance running by providing a heart rate control mechanism using auditory feedback, more precisely, one can select between a metronome or music.

A user can specify the desired heart rate evolution before starting a workout, in terms of heart rate over time. During the workout, the internal proportional controller uses the current and desired heart rate, as well as the current step frequency to calculate the appropriate next beats per minute of the playback. This entire process is based on the assumption, that the user adapts his body movement (expressed by the step frequency) to the bpm of the playback and the movement itself is closely tied to the heart rate. Figure 2.2 and 2.3, show screenshots of the BeatRunner app before starting a workout and during a workout, respectively.

As previously discussed research by Cavagna et al. [15, 16] showed, the freely chosen step frequency during running is nearly constant for speeds below 10 km h^{-1} , which complicates the approach of using the bpm of an audio signal to control the heart rate for those particular slower speeds. In our Endomondo data set (see Section 4.2.3), the average speed of all running workouts was $10.7 \pm 2.6 \text{ km h}^{-1}$. Therefore, many workouts lie in the area below 10 km h^{-1} where the

²A sports trainer for a Swiss cantonal regional orienteering elite.

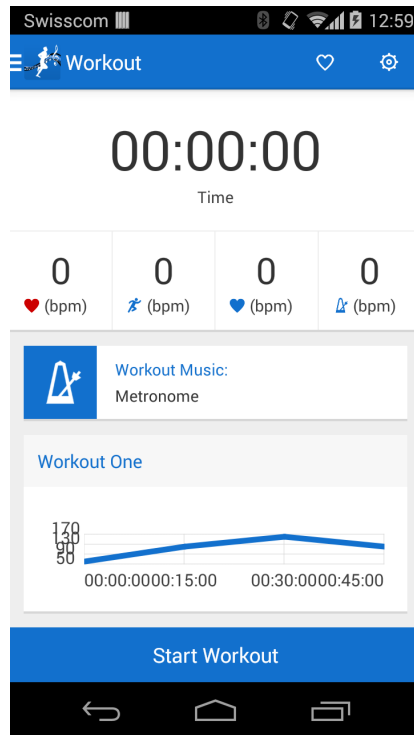


Figure 2.2: BeatRunner workout configuration screen, where the desired heart rate over time and the kind of audio signal (music or metronome) are specified.

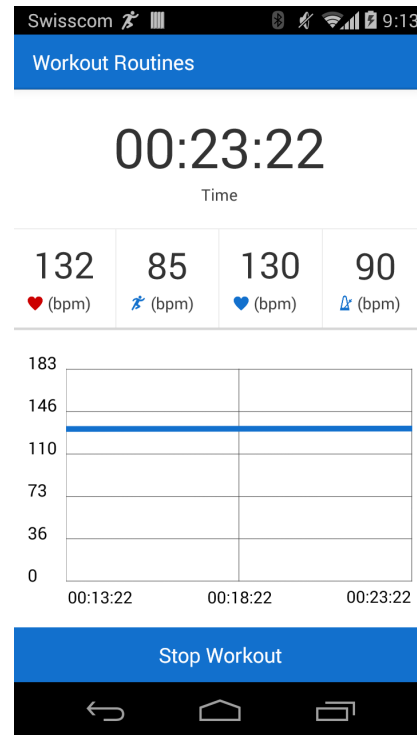


Figure 2.3: BeatRunner during a workout session, displaying current values for heart rate and step frequency, as well as desired heart rate and bpm of the playback.

freely chosen cadence is typically near constant. Even though the beat of a song may influence the runner slightly in his step frequency, runners typically do not feel comfortable when running with a different step frequency than the one they would naturally choose for a given speed. This is one of the reasons, why this thesis does not focus mainly on improving the BeatRunner system, but rather on finding well performing heart rate and fitness model for running workouts.

We used BeatRunner in our thesis as a foundation for our data collection process. Additionally to the already present logging capabilities, we added:

1. GPS location and speed tracking
2. A better step frequency algorithm (see Section 4.1.1)
3. Raw logging of accelerometer, gyroscope, barometer, and other sensors
4. Stride sensor support

5. Automatic upload of the workout data to a webserver when a Wi-Fi connection is available

In Chapter 7, we discuss how our findings could be applied to improve the beat selection of BeatRunner.

Methodology

The main goal of this thesis is to model the heart rate during exercise. This is done by observing the main influences on the energy consumed during a workout session. As the heart rate gives a good indication of the energy consumed, we can establish relations between said influences and the heart rate.

In Section 3.1 we discuss and investigate different factors that influence the heart rate significantly. We then establish methods to approach building a heart rate model based on two data sets of workout data we gathered. Chapter 4 contains details about the gathered data and performed processing steps.

Based on the heart rate model, Section 3.3 addresses methods to deduct a user-specific fitness measure to allow a user to track his fitness level progress. In Section 3.4, we show how the built models and measures are evaluated.

3.1 Influences on Heart Rate

While there are various factors influencing the heart rate [36], we concentrate on the ones corresponding to the limbic system, more precisely the effects related to physical exercise. Those can often be measured directly during a workout session and then brought into relation to the heart rate.

We distinguish two groups of influences on the heart rate during exercise:

Direct Influences

Everything directly related to the body of the subject. They can usually directly be influenced by decisions of the subject.

External Influences

Influences not directly affected by the subject, resembling the environment around the subject.

We list some main influences for both categories in the following sections.

3.1.1 Direct Influences

Speed

Increasing speed leads to a higher heart rate, while lower speeds relax the body and the heart rate decreases.

Cadence

The cadence or step frequency has a close tie to speed but also to step length. It also is a direct influence on the heart rate during workout sessions both in running and cycling (where it corresponds to the number of revolutions of the crank per time).

Clothing

Although this has only a minor influence, the type of clothing worn by a subject certainly affects the energy consumed. Typically, light clothes lead to lower energy consumption. This includes the choice of shoes.

Fitness

Fitness is not controllable by the subject from one workout session to the next and is rather something that changes over longer periods of time. It clearly is an essential influence on the subject's heart rate.

Daily Condition

Each day is different and there are natural variations in the daily health. Sometimes we feel fitter and sometimes more tired. This has a significant influence on the performance one is able to deliver. We also consider fatigue to be an element of the daily condition.

Technique

Be it the footstrike technique of a runner or the way a cyclist shifts her or his weight when going up and down, there are several factors influencing the effort of an athlete that come down to technique and experience.

Person

There exist general differences between multiple subjects but also influences based on weight, height, age or gender.

3.1.2 External Influences

While external influences cannot trivially be changed by a subject directly, they can often be controlled. For example a person going for a run can choose not to go outside when it's raining or take a flat route with few slopes.

Grade

A well studied influence on the energy consumed during exercise. Sec-

tion 2.1 contains more details and discusses some studies conducted by sport and health care researchers.

Path

Both the general track condition as well as the surface on which a workout takes place influence the energy consumption.

Weather

Several weather related factors influence the effort needed during an exercise session. Some, like temperature, humidity, and wind can be quantified, while others can merely be labeled (rainy, foggy, light clouds...).

3.1.3 Measurement

We gathered a lot of workout data from multiple sources, which contains measurements of some of the above mentioned influences on the heart rate during exercise. In Chapter 4, we discuss this process more thoroughly.

The difficulty to measure some of the mentioned influences on the heart rate, directly limits the precision and accuracy a heart rate model can achieve.

3.2 Heart Rate Model

We aim at building a model that takes the main features of a workout session as an input and can give accurate heart rate estimates with as little observations as possible.

Based on gathered workout data, we can train a model with real heart rate data and use the trained model to estimate the heart rate in cases where no heart rate is present or evaluate the model in comparison to the measured heart rate.

Based on the literature research (Section 2.1), we decided to focus on the following workout features for an initial heart rate model:

- Speed [m s^{-1}]
- Cadence [min^{-1}]
- Grade [m s^{-1}]

We note, that the grade is measured as the vertical speed rather than a fraction of vertical and horizontal distance. The vertical speed can be measured more precisely as we do not rely on any speed measurements from a GPS or

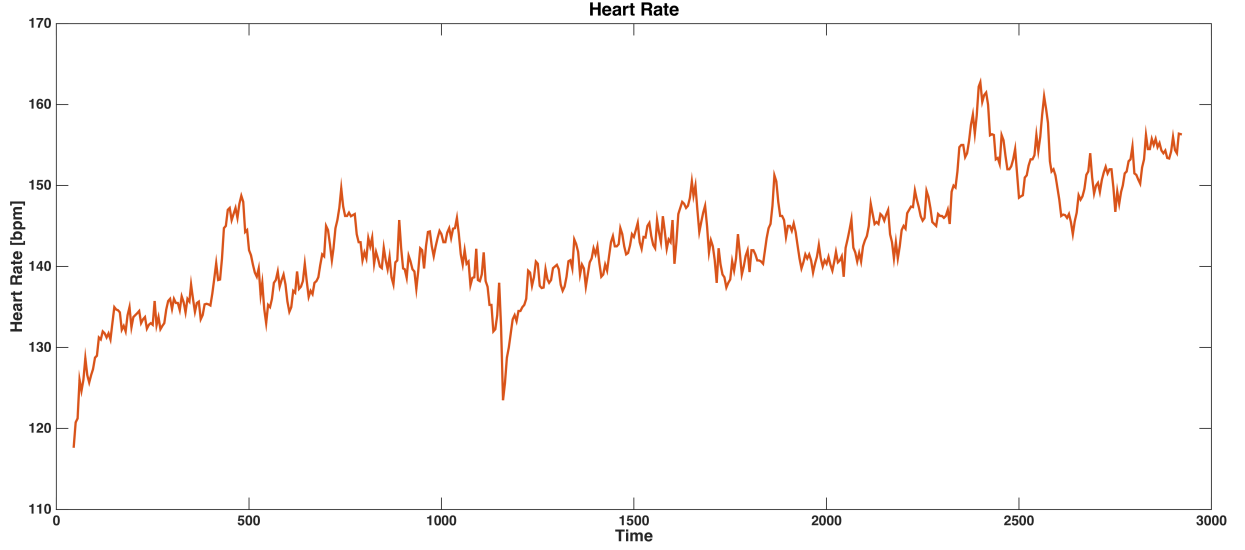


Figure 3.1: Heart rate visualized over time during a workout session with near constant intensity except for some small slopes. We notice that the heart rate tends to increase steadily by a small amount.

similar source. Instead we only consider elevation data (from a barometer or external source) and a time interval (See Section 4.1.1 for more details).

When examining some heart rate data from a workout session, like the one seen in Figure 3.1, we notice that the heart rate has a trend to slowly increase over time without significantly increased intensity (speed and cadence not shown). This effect is called cardiovascular drift and has been studied before [37, 38]. It is influenced by internal body temperature, hydration, and the amount of active muscle mass during exercise. Our heart rate models take cardiovascular drift in consideration to achieve a better fit.

Another interesting observation regarding the relation of our key workout features and the heart rate is the delay between a change of the workout intensity and a corresponding change of the heart rate. We observed such delays in the area of 15 s to 30 s. Section 5.1.2 establishes a method of considering this delay in our heart rate models.

3.3 Fitness Measure

The literature suggests multiple ways how the general fitness of a subject can be measured and how changes in fitness can be detected. (See Section 2.4)

We developed a unique way of modeling fitness based on our heart rate model. Under the assumption that the fitness level of a subject gradually increases with

regular exercise, we train the model for different subsets of a subject's workouts and investigate changes in the learned model parameters over time. This can give a relative indication when the fitness level changes. The main purpose of the fitness measure is to accurately reflect an improved fitness which could be used to motivate a potential user.

3.4 Evaluation

Apart from a training and a test set, we established a non-overlapping evaluation set that we solely use to evaluate our heart rate models and fitness measures.

Since the heart rate model is trained per subject, the evaluation set consists of workouts from multiple subjects that have not been used for testing before. A part of those workouts are then used to train user specific models, while the rest is used for evaluation by measuring the estimation error.

The whole heart rate model building process is therefore split in the following parts:

1. Train model per subject on a fraction of available workouts
2. Test model based on remaining workouts of subject
3. Improve model in several iterations
4. Evaluate model based on previously unused data from other users

Chapter 6 contains the detailed discussion of the evaluation and its results, while Chapter 5 explains the models in more detail.

For the fitness model, the evaluation is strongly based on the assumption that the fitness level of a subject increases over multiple workout sessions. Section 6.2 proposes a improvement function and tests its output for different configurations. As there is no ground truth available (we do not know the real fitness level of a subject), the evaluation can only give limited information about the performance of the proposed model. We discuss some literature in Section 5.3 to motivate the choice of components for the fitness measure.

Data Sets

This chapter describes the workout data sets used to train, test, and evaluate both the heart rate and fitness models as described in Chapter 5. The data sets consist of various features from workout sessions by multiple test subjects performing sports like jogging/running, mountain biking and cycling.

In a first step, we collected data from multiple workout sessions individually, using Android smart phones. Most features could be recorded with the built-in sensors, while some required extra hardware that was connected via Bluetooth. Section 4.1 contains details about the individual data collection process, the conducted processing and a short evaluation about the data quality and its use for later tasks.

Because of limited hardware availability due to the comparably high cost, we also used workout data freely available on the internet as a second data set. The selection of a suitable data provider, the collection of the data and further details about the data set itself are described in section 4.2.

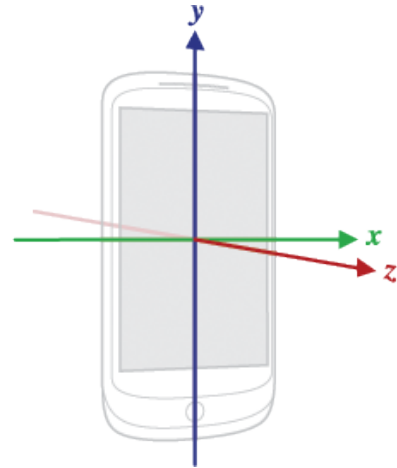


Figure 4.1: The three device axis for Android phones. [39]

4.1 Individual Collection

To investigate various influences on the heart rate during a workout session, we collected data from as many sensors as possible, namely the following:

Accelerometer

This sensor reports a three-dimensional acceleration vector with each value

in m s^{-2} , describing the forces currently acting on the phone. In the device's default orientation, the axes are horizontal to the right, vertical pointing up and pointing out of the front face of the screen for X, Y and Z respectively as shown on Figure A.1.

Gyroscope

The gyroscope gives information about the current orientation of the phone. Every sensor value contains the rate of rotation around each axis in rad s^{-1} .

Linear Accelerometer

As a so called *virtual sensor*, the Linear Accelerometer is not an actual hardware sensor, but gets its value through combining data from multiple hardware sensors. In this case, it uses both the gyroscope and the accelerometer to provide an acceleration vector that does not include the effect of earth's gravity.

Gravity Sensor

Also a virtual sensor, that reports the gravity in m s^{-2} acting on the phone. The values are equivalent to the difference between the acceleration and the linear acceleration.

Barometer

The barometer has yet to become a standard sensor for mobile phones as only a few newer models have one built in at the time of writing. It measures the current atmospheric pressure in hPa, which can be used to calculate the elevation of the phone. (see Section 4.1.1)

Heart Rate Monitor

We used Bluetooth enabled Heart Rate chest belts¹ that can be connected to the phone and be used to get regular updates about the current heart rate of the subject in beats per minute. Figure 4.2 shows one of the chest belts used for data collection.

Global Positioning System (GPS)

Each data point consists of a timestamp, a longitude, a latitude, an altitude, the instantaneous speed and an accuracy in meters². They were collected with highest available precision in regular intervals not exceeding 5 seconds.³

Stride Sensor

A Stride Sensors can be attached to a runners shoe and can measure both

¹Polar H7 Heart Rate Sensor

Available at: http://www.polar.com/en/products/accessories/H7_heart_rate_sensor [Accessed 16.04.2015]

²The accuracy is defined to be the radius of 68% confidence [40]

³The system decides when exactly to provide a new location based on different factors like movement or speed

stride length and step frequency with high accuracy. It transmits the data regularly to the phone using Bluetooth. We used it to have a comparison to our own step frequency algorithm that is described in more detail in section 4.1.1.

Each sensor value comes with a timestamp which is either directly reported from the sensor API or manually set when the app receives the notification of a value change. In both cases, the reported value is in nanoseconds since the last device restart and can easily be converted to the time since the start of the workout.

Additionally, we manually collected basic information about each test subject and the environment the workout took place in:

- Subject Information
 - Gender
 - Age
 - Height [cm]
 - Weight [kg]
- Environment Information
 - Temperature [°C]
 - Humidity [%]
 - Time of Day
 - Road Surface

In a further step, some of the raw feature values were processed to find core characteristics of running sessions. Examples for those are the instantaneous cadence from the accelerometer data or the grade from elevation data, which in turn is deducted from the barometric pressure. The following sections explain the processing steps performed on the raw data sets, the evaluation of the data set and individual features, as well as some general information about the gathered data.

4.1.1 Method

Most features were directly recorded using the smart phone app BeatRunner. It was extended to log all the sensor data mentioned above in a simple Comma-Separated Values (CSV) format. Environmental information like the outside temperature and humidity were measured using a consumer grade weather station and the road surface was annotated manually by the subject.

We describe the workout setup and the post processing steps in the following sections.

Setup



Figure 4.2: A heart rate monitor used for heart frequency collection.



Figure 4.3: Smart phone attached to a subject's upper arm. Also visible is the stride sensor on the subject's right shoe.

In an attempt to calculate the instantaneous step frequency of a subject during a running session, the placement of the phone (and therefore the accelerometer) is crucial. During walking, jogging, and running the entire body undergoes a periodical vertical up and down movement for each step taken. For all conducted training sessions, we attached the phone on the upper arm using an off-the-shelf armband, to ensure it can capture the body movement as precisely as possible without additional noise due to possible loose attachment. Figure 4.3 shows the attached smart phone on the upper arm of a test subject.

Speed

We investigated two possible approaches to determine the speed of the subject during a training session:

GPS Speed

The GPS does directly report a speed value with each location. Some devices do this by analyzing the Doppler Shift of the GPS signal while

others use the distance to previous locations. [41] However, this value is relatively inaccurate and often not even present due to low signal quality. [42]

GPS Locations

Instead of relying on the directly reported speed values from the GPS, we can calculate the speed manually from the last few locations.

For our test set, we calculated the distance between successive locations (on basis of the WGS 1984 reference ellipsoid) and used a rolling average over 5 locations to get an approximation of the instantaneous speed of the subject.

Elevation

The energy cost of running uphill and downhill slopes has been investigated extensively. [8, 9, 10] The grade has a major impact on the total energy consumed and as a result has a direct impact on the heart rate of a subject. It is therefore essential to capture the grade as a feature of an expressive heart rate model.

The grade itself is hard to measure directly using a smart phone. Even positional sensors cannot help in this case, since the subjects stance does not accurately reflect the grade of the ground. Additionally, positional measurements during user movement are very noisy.

We analyzed and compared two ways of retrieving the current elevation:

1. Elevation reported by the GPS
2. Atmospheric pressure reported by barometer sensor

The value reported by the GPS is the altitude in meters above the WGS 1984 reference ellipsoid, a reference surface defining the nominal sea level as specified in [43]. It is however very inaccurate, as a comparison to Google Earth data showed an average accuracy worse than 20 m with a precision of less than 10 m.

On the other hand, the barometer can detect even minimal changes in atmospheric pressure, corresponding to elevation changes of as little as 1 m. We can transform the barometric formula for constant temperature and humidity [44] as seen in Equation (4.1), to retrieve a version that can be used to calculate the altitude over sea level from a given pressure value, yielding Equation (4.2).

$$P_z = P_0 \cdot \exp \left[-\frac{g \cdot M \cdot z}{R \cdot T} \right] \quad (4.1)$$

where

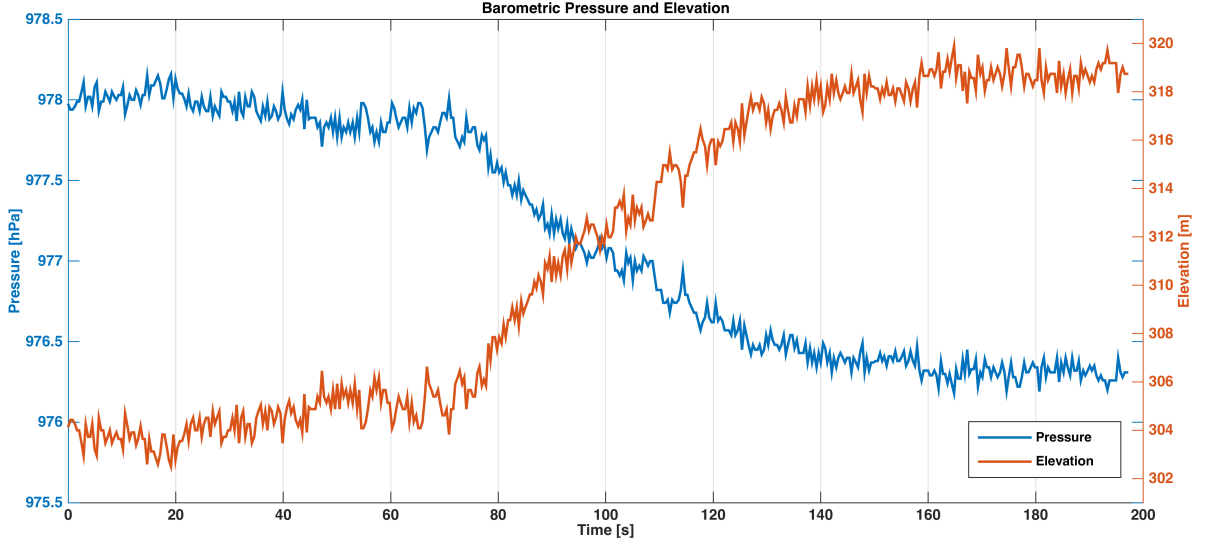


Figure 4.4: Barometric Pressure measurement and calculated elevation (using Equation (4.3)) during a short ascent of approximately 16 m during a time of around 90 s.

P_z : static pressure at height z [Pa]

P_0 : static pressure at height 0 [Pa]

z : altitude [m]

T : temperature [K]

g : gravitational acceleration on earth surface, i.e. 9.80665 m s^{-2}

M : molar mass of dry air, i.e. $0.0289644 \text{ kg mol}^{-1}$

R : universal gas constant, i.e. 8.31447 J/(molK)

$$z = -\log \frac{P_z}{P_0} \cdot \frac{R \cdot T}{g \cdot M} \quad (4.2)$$

When P_0 is set to the atmospheric pressure at sea level (i.e. 1013.25 hPa), z is the elevation over sea level. For our application we are ultimately interested in calculating the grade. This means we want to know the change of elevation.

As a varying temperature T only has a minimal effect on the final result when calculating elevation differences, we assume it be constant between two measurements (less than a second apart), and set it to $T = 20^\circ\text{C} = 293.2 \text{ K}$. Equation (4.3) shows the final equation after substituting all constants. The pressure and calculated elevation for a short uphill slope with an ascent of approximately 16 m during a workout session, can be seen in Figure 4.4.

$$z \approx 8582.49037 \cdot [11.52609 - \log P_z] \quad (4.3)$$

Grade

The grade g (or slope) is a measure of incline and can be expressed as the ratio of elevation difference Δz (vertical distance) to run d , the horizontal distance. This relation is shown in Equation (4.4).

$$g = \frac{\Delta z}{d} \quad (4.4)$$

Our elevation data consists of tuples (t, z) with a timestamp t and a corresponding elevation z , the horizontal distance cannot be directly deducted. Therefore distance information from the GPS would have to be used to allow for a calculation of the grade.

Since the instantaneous speed as well as short distance measurements from the GPS show significant noise, we decided to not adapt the exact definition of the grade and rather consider the horizontal distance per time (instead of per horizontal distance). We end up with the instantaneous vertical speed of the subject in ms^{-1} - calculated based on Equation (4.5).

$$v_{\text{vert}} = \frac{\Delta z}{\Delta t} \quad (4.5)$$

In consideration of the energy consumed by the subject's body, this measure reflects a direct influence as the energy is proportional to the actual speed one climbs or descends a slope rather than only its grade.

Step Frequency

The step frequency, or cadence is the number of times a foot of a subject touches the ground during a given period of time. It is commonly measured in min^{-1} . It is distinguished from the stride frequency, which describes the number of strides (i.e. both feet touching the ground once) per time.

Given the data from the accelerometer, it is possible to detect every step taken, since the subject's body is in a periodic up and down movement during running. Coupled with timing information, it is possible to deduct the step frequency from the detected steps. A device which performs this task is called a pedometer.

To account for possible changes of the phone's orientation during a workout, we considered the total length of each acceleration vector $\|A\|$, as calculated using equation (4.6), rather than its individual components.

$$\|A\| = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (4.6)$$

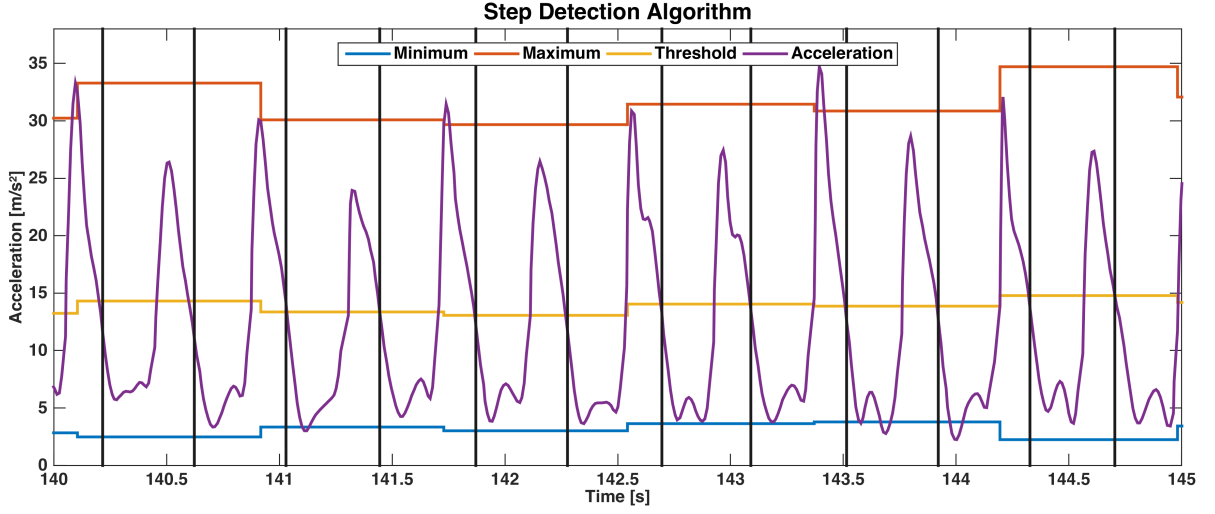


Figure 4.5: 5 s extract of the total acceleration during a training session showing the step detection algorithm in action. A step is detected when the acceleration (purple) crosses the dynamic threshold (yellow).

Figure 4.5 shows the total acceleration in m s^{-2} as a purple line over a period of 5 s during a training session.

Our pedometer algorithm is inspired by [45] and uses a variable threshold (yellow) that has to be crossed from top to bottom by the acceleration value for a step to be detected (black lines). The threshold is the weighted average of the maximum (weight 0.45; shown in blue) and minimum (weight 0.55; shown in red) values, the total acceleration reaches in windows of 80 samples. The used phones produced around 90 to 100 samples per second, which makes each window a bit shorter than one second. This ensures, that at least one entire period of the accelerometer oscillation (one step) is included and therefore true extrema are detected.

To account for noisy data and some edge cases, the algorithm only detects steps that have a length between 0.3 s and 1.2 s and requires at least 4 successive valid steps before starting to produce new cadence values.

The final step frequency produced by the algorithm is the average of the last 4 steps detected, which helps reducing some noise in the data while keeping updates reasonably frequent. An example can be seen in Figure 4.6.

In Table 4.1, the configuration values for the algorithm are listed. The pseudo-code in Algorithm C.1 shows a possible implementation of the described step detection and cadence calculation algorithm. Although the listing receives the entire acceleration data set at once, the algorithm does work in an online environment with a delay of 4 steps (the rolling average window used for cadence calculation).

Variable		Value	Comment
Step Detection			
Threshold Sample Window	τ^w	80	Number of samples considered for threshold calculation
Threshold Maximum Weight	τ^{maxw}	0.45	Weight for Maximum; Minimum weight = $1 - \tau^{maxw}$
Step Filtering & Cadence Calculation			
Maximum Step Length	Δs^{max}	1.2 s	
Minimum Step Length	Δs^{min}	0.3 s	
Cadence Start Step Count	c^{update}	4	Successive valid steps to update cadence
Cadence Reset Step Count	c^{reset}	10	Successive invalid steps to reset cadence to 0
Step Window Size	w	4	Steps considered for cadence calculation (rolling average)

Table 4.1: Configuration values of the step detection algorithm.

The implementation used during our experiments is slightly more complex than the algorithm shown in here. It additionally includes:

1. A minimal sample count required for a step to be valid (8 samples)
2. A minimal absolute change (0.02 m s^{-2}) in acceleration needed for a data point to be included in calculation. This reduces some high frequency noise.

We also tested out a restriction of the allowed change in step length, but this did not significantly reduce noise and was therefore not used in the final version of the algorithm.

4.1.2 Evaluation and Filtering

To evaluate and ensure the quality of the gathered data, we compared some features with other alternative measurement methods and applied some basic filtering processes to limit noise.

Step Frequency

We used a stride sensor by Polar⁴ as a comparison to evaluate the step frequencies calculated from the accelerometer data. The stride sensor can be connected

⁴Polar Stride Sensor Bluetooth Smart

Available at: http://www.polar.com/en/products/accessories/stride_sensor_bluetooth_smart [Accessed 16.04.2015]

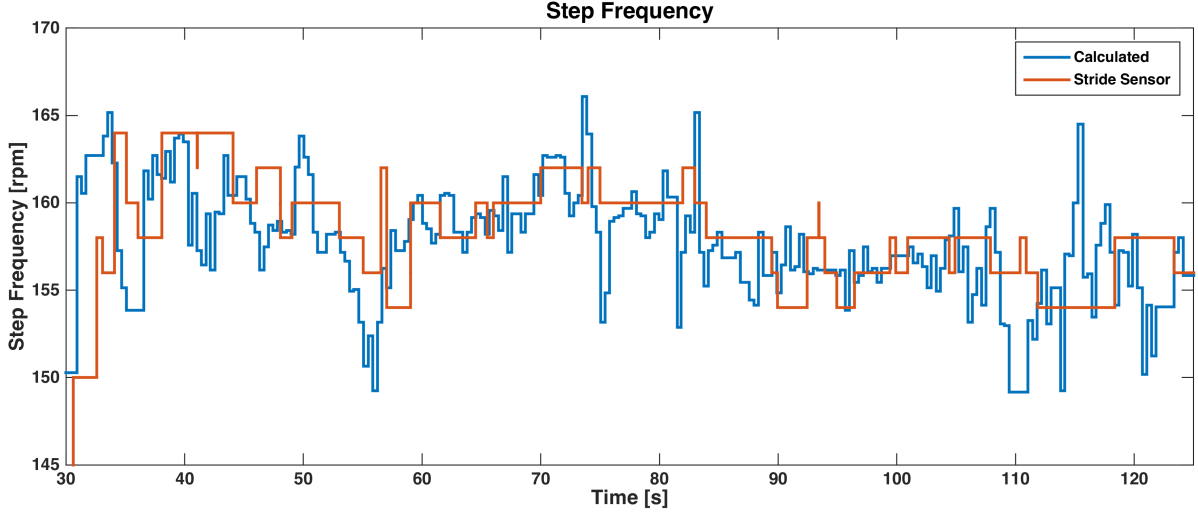


Figure 4.6: The calculated cadence during part of a workout session in comparison to the values reported by the stride sensor.

to a smart phone using Bluetooth Low Energy. It uses a standardized protocol, defined in the *Running Speed and Cadence* service and characteristics specifications for Bluetooth Generic Attribute Profile (GATT). [46, 47]

Figure 4.6 shows a short part of a training session, comparing the calculated cadence with the one reported by the stride sensor.

Although neither our algorithm nor the stride sensor provides an actual ground truth, we can assume, that the stride sensor delivers more accurate results since it has an optimal position on the foot of the subject. While we tested some different configurations of the step frequency algorithm, we minimized the deviation from the stride sensor data. Figure 4.7 shows boxplots of the deviations for different values of the maximum weight threshold τ_{maxw} . We found, that a value slightly below 0.5 seems to produce better results and therefore our final algorithm uses a value of $\tau_{maxw} = 0.45$. We also made some further manual tests of different configuration values during development but don't show the details here.

We also notice, that the stride sensor delivers significantly less updates than our step frequency algorithm. Table 4.2 shows, that the mean interval for the algorithm is about 0.4 s which corresponds to an average step since the algorithm updates the cadence on every valid step (given the previous $c^{update} - 1 = 3$ steps were valid as well).

When considering an average running session, we won't expect the step frequency to change rapidly very often. Therefore it is possible, that a more conservative cadence update frequency could even help reducing some noise in the

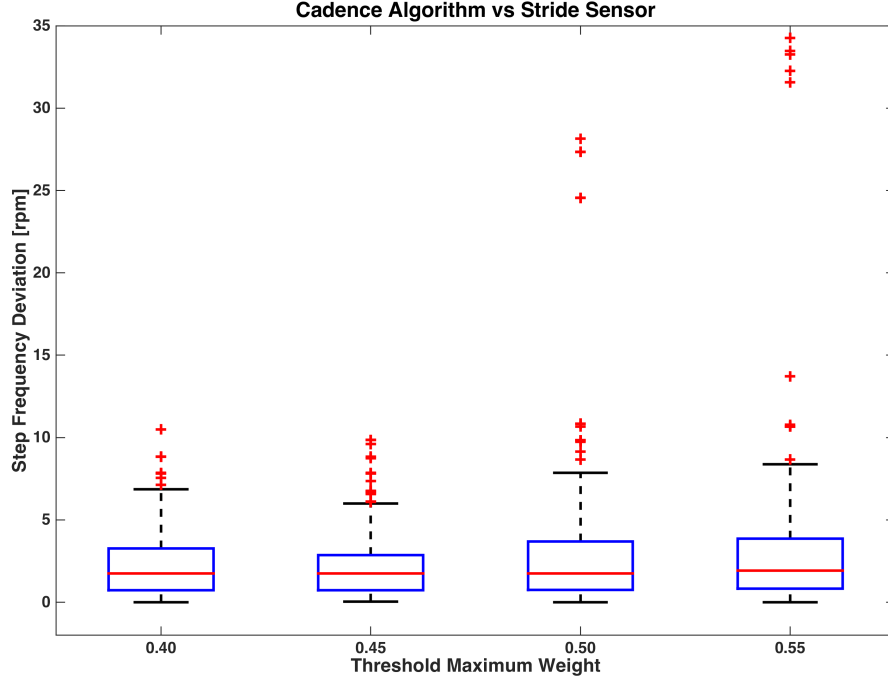


Figure 4.7: Boxplot showing the absolute deviation of stride sensor data and calculated step frequency for different weight thresholds τ^{max} in the step frequency algorithm. Since a weight of 0.45 showed the best results, this is the value selected for the final algorithm. Detailed numbers for it can be seen in Table 4.2.

output. We did however not test this and can just make assumptions based on the deviations we observed in plots like the one seen in figure 4.6.

Hardware

To gather workout data for the individual collection, we used a total of three different Android smart phones:

1. Motorola Moto X (1st Generation)
2. LG Nexus 4
3. Samsung Galaxy S4 mini

The Samsung phone however, does not include a barometer sensor and did not contribute any workout instances in the final collection. The other two phones were mainly used to compare the sensor measurement quality, especially of the GPS. We could however not determine any significant differences in accuracy or precision.

Property	Algorithm	Stride Sensor
Time between values	$\mu = 0.41$ s	$\mu = 1.01$ s
	$\sigma = 0.18$ s	$\sigma = 0.61$ s
Deviation	$\mu = 2.77$	
	$\sigma = 8.76$	
	$\sigma_{1/2} = 1.76$	

Table 4.2: Comparison of calculated cadence and stride sensor data during a 42 min workout. μ denotes the mean, σ the standard deviation, and $\sigma_{1/2}$ the median deviation.

4.1.3 Data Set Properties

In our individual collection, three subjects (one female, two male) of ages 22 ± 2 , carried a smart phone running the BeatRunner application while running.

Unfortunately, in some cases there were complications, which made some captured workouts unusable.⁵

When we concluded, that the number of workouts we could record this way is too low as sole data source to build a heart rate model, we concentrated on the online data set described in the next section.

4.2 Online Data

Apart from the individual data collection, we also looked into the data found on online fitness platforms. Nowadays there are many widely used smart phone applications available that offer different fitness tracking features. While some include only basic functionality to tell a user how far and how fast she or he ran during a jogging session, others include an entire social platform where workout data is made available online for other people to see. The possibility to compare own workouts with those of friends, people running similar tracks or even fellow runners from the other side of the planet, is for many runners an incentive to train even more.

While privacy preferences of users limit the number of publicly visible workouts, larger services still have thousands of training sessions available for everyone to see. To find a suitable data provider, we compared the following three services:

RunKeeper [www.runkeeper.com]

RunKeeper focuses on providing training routes from other athletes with associated workout details as an incentive for users to try a particular route

⁵One subject had his GPS disabled, another one had problems pairing the provided heart rate belt with the phone

for themselves. The full workout data is however only available to premium users. Normal users just see some general statistics like the average pace or heart rate rather than multiple values over the entire duration of a workout session.

Runtastic [www.runtastic.com]

With more than 10 million downloads in the Google Play Store, the Runtastic Running app⁶ is one of the most downloaded fitness applications for smart phones. It measures key factors like heart rate (with a compatible chest belt), pace, and location while running. Similar apps for other sports also exist from the same developer. Unfortunately, details about other user's workouts are only fully visible when purchasing a premium account.

Endomondo [www.endomondo.com]

Endomondo is an entire social fitness network. Users track their workouts and fitness using a mobile application and website. Many of them make their information publicly available. Figure 4.8 shows a screenshot of a mountain bike workout on the Endomondo website. Apart from basic statistics like duration, weather, calories burned and average heart rate, it also includes a graph of speed, heart rate, cadence, and elevation over the entire workout session.

All three services above offered a rich set of workout data but not all data was accessible without premium user accounts. We therefore decided to use Endomondo as our primary source for the online workout data set. There are multiple million workouts on the website of which a fair fraction is publicly accessible. To qualify as input for model training, a workout has to contain heart rate data over its complete duration; which is available on many Endomondo workouts.

In the following sections, we describe how we retrieved the workout data and what processing steps were performed to ensure sufficient data quality. Section 4.2.3 includes statistics about the workouts generally available on Endomondo and especially the subset which is contained in our online data set.

4.2.1 Method

Every workout on the Endomondo platform is identified by a unique `id` which is part of an URL leading to the associated workout data. Such an URL is of the form:

⁶Runtastic Running & Fitness in the Google Play Store
Available at: <https://play.google.com/store/apps/details?id=com.runtastic.android&hl=en>
[Accessed 19.04.2015]

⁷Endomondo Workout 335607570
Available at <https://www.endomondo.com/workouts/335607570> [Accessed 15.04.2015]

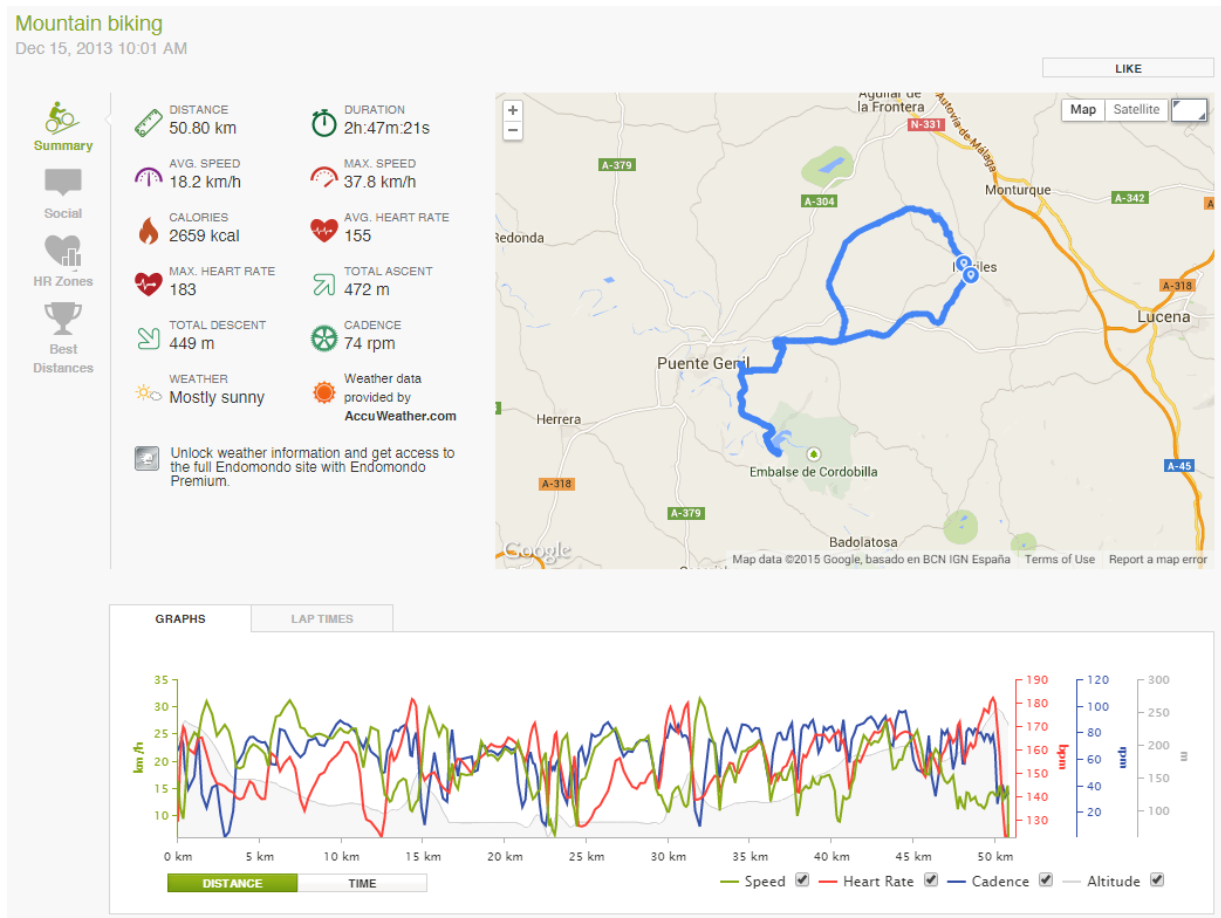


Figure 4.8: Screenshot of a mountain bike workout on the Endomondo website including information like duration, distance, weather and coordinates visualized on the map as well as speed, cadence, heart rate and elevation visualized in a graph.⁷

<https://www.endomondo.com/workouts/<id>>

To automatically retrieve workout data from Endomondo, we programmed a website scraper to iterate over possible ids and evaluate if the workout is public and contains the desired features. Most importantly, it has to contain heart rate data to be eligible for our data set.

Scraper

For the scraper implementation we built a python script using the libraries `requests`⁸ for HTTP request handling and `beautifulsoup4`⁹ for HTML parsing and DOM like access on the webpage source.

Many workout features are directly available in the HTML source code of the corresponding website. Coordinates and graph data however, have to be extracted from an inline JavaScript asset in JSON format. The format of the part containing coordinates, heart rate, cadence, speed, altitude, distance, and time measurements, is shown in Appendix D.1.2.

During the scraping process, we analyzed the source code of the workout website to ensure it contains heart rate data and only then downloaded all available metadata together with the data used in the plots in JSON format. In a second step, we modified the scraper to find and analyze all workouts of a given user, identified by his or her unique Endomondo `user-id`. This led to the scraping process being divided into two main parts:

1. Iterate through `id` starting from 1 and download all eligible ones
2. For each downloaded workout, find all other eligible workouts by the same user

We downloaded 210 614 workout sessions from 3272 different users in total, accounting for an average of $\sigma = 64$ workouts per user. The scraping process was significantly slowed down because of a query limit for Endomondo pages. In average we were able to download less than two workouts per second, which significantly reduced the total number of workouts we collected.

Filtering

Figure 4.9 shows the distribution of different sports in all 210 614 downloaded workouts. However, we are only interested in workouts in one of the three sports running, cycling, and mountain biking. Additionally, we want to be sure to have cadence data available in our workout data set, as the cadence has a major impact on the energy consumed.

Many of the downloaded workouts were missing a small fraction of heart rate measurements or contained other minor data corruptions. In a further filtering step, we selected the workouts we wanted to use as our training and test sets for

⁸Official website of python `requests`

Available at: <http://docs.python-requests.org/en/latest/> [Accessed 19.04.2015]

⁹Official website of the python `beautifulsoup4` library

Available at: <http://www.crummy.com/software/BeautifulSoup/> [Accessed 19.04.2015]

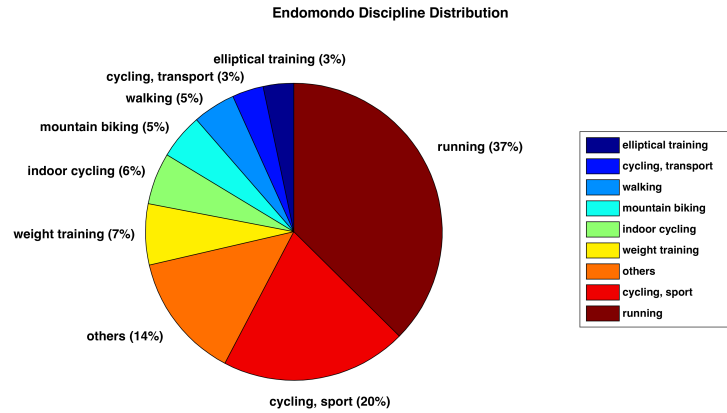


Figure 4.9: Distribution of different disciplines in the retrieved Endomondo data set with a total of 210 614 workout instances.

heart rate modeling. Figure 4.10a shows, that in the end, less than 1 % of all downloaded workout instances were selected for our filtered set. Most workouts were ineligible because of either missing cadence values (49 %) or wrong sports (37 %). Some also had missing or corrupt altitude, location or heart rate data.

What is left over are 2199 workout instances distributed over the three sports running, cycling, and mountain biking as seen in figure 4.10b.

Structured Collection

In our data collection, all workout instances are grouped by user in a user-specific folder named with his or her `user-id`. Additionally to one JSON file per workout, each folder contains a `profile-<user-id>.json` file which includes all meta data that was available on the user's Endomondo profile. Dependent on a user's privacy settings, this includes gender, height, weight; sometimes even birthdates, countries or postal codes. Apart from the personal information, it typically also contains general statistics about all Endomondo workouts like the best performances for a certain distance in each sport or the total calories burned.

4.2.2 Data Quality

The Endomondo elevation data is based on the GPS altitude, which is quite inaccurate and imprecise (as described in Section 4.1.1). Therefore we used the Google Elevation API [48] to retrieve better elevation data along the waypoints of the workout tracks.

Although the Elevation API can not accurately reflect structures like bridges

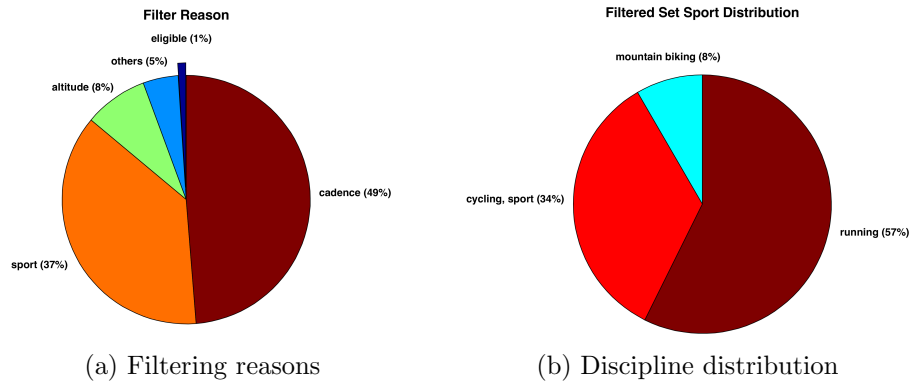


Figure 4.10: Statistics on the filtering process of the collected Endomondo workouts. After filtering, a total of 2199 workout instances remain. (less than 1 %) Only running, cycling, and mountain biking are part of the filtered data set.

or small and short slopes, it still provides both better precision and accuracy than the GPS based data from Endomondo.

4.2.3 Statistics

The final Endomondo workout collection, builds two data sets:

1. Complete Collection
2. Filtered Collection

While the complete collection contains all downloaded workouts and does not come with any special guarantees, all workouts in the filtered collection are ensured to:

- be from a sport in {running, cycling, mountain biking}
- contain heart rate data over time
- contain cadence data over time
- contain GPS location data
- contain Google Earth altitude data (Elevation API; see Section 4.2.2)

Table 4.3 shows statistics about the two Endomondo data sets in direct comparison. For the filtered data set, Figure 4.11 shows the distribution of workouts per user. Unfortunately 56 % of all users, have less than 6 workouts that fulfill

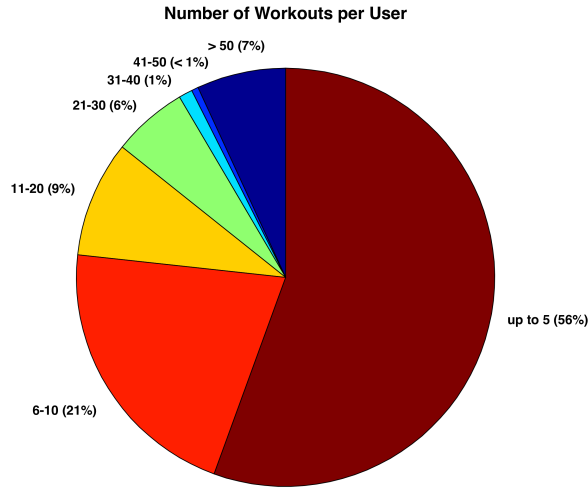


Figure 4.11: Distribution of number of workouts per user. 56 % of the 189 users have five or less workouts while some 13 (7 %) have more than 50 workouts.)

all conditions listed above. For model training, users with more workouts are more suitable since the heart rate model is trained and tested per user.

An interesting measure is the time between two observation values in the Endomondo data set, since it is not very consistent. Workouts have some 287 ± 44 data points, largely independent of their duration. This leads to a large distribution of the mean time between observations, which is shown as a histogram in Figure 4.12. Most workouts have a mean time between values below 30s, which still makes them reasonably dense for use in a heart rate model.

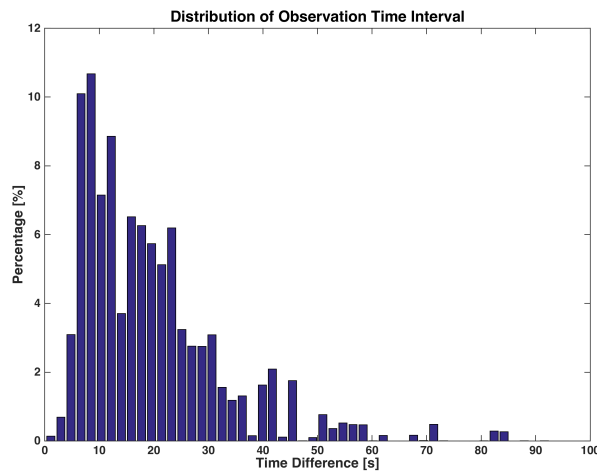


Figure 4.12: Distribution of time between values for workouts in the filtered Endomondo data set.

Property	Complete	Filtered
General Statistics		
Number of Workouts	210 614	2199
Number of Users	3447	189
Number of Disciplines	57	3
Workout Statistics		
Number of data points	n/a	287 ± 44
Time between values [s]	n/a	20 ± 18
Average Speed [km h^{-1}]		
Running	n/a	10.7 ± 2.6
Mountain Biking		21.5 ± 8.8
Cycling		20.8 ± 18.2
User Statistics		
Gender [σ / φ]	2805 / 443	174 / 6
Age [years]	40 ± 10	42 ± 9
Height [m]	1.77 ± 0.09	1.80 ± 0.07
Weight [kg]	80 ± 19	79 ± 10
BMI [kg m^{-2}]	24.4 ± 3.0	24.3 ± 2.4
Number of Workouts	443 ± 408	816 ± 616
thereof		
in Collection	62.6 ± 143.7	12.3 ± 26.5
in Collection [%]	16.5 ± 27.4	2.1 ± 4.9

Table 4.3: Statistics for the complete and filtered collection of scraped Endomondo workout session data. Numbers are means and standard deviations; when information was unavailable, the subject or workout was ignored. The complete statistics are listed in Appendix D.1.3.

Heart Rate Model

In this chapter we describe the heart rate models we built and show how each of them can be applied to the collected data sets. In Section 5.3 we show, how the model based on a Conditional Random Field, can be used to make statements about a subject's fitness level.

The first section, 5.1, explains the preprocessing steps we performed on our raw data sets, in order for them to be usable in our models.

The main workout features we selected for modeling the heart rate are:

1. Speed v [m s^{-1}]
2. Grade; more precisely, the vertical speed g [m s^{-1}]
3. Cadence c [s^{-1}]

In Section 5.2, we introduce three types of heart rate models, based on three different mathematical approaches:

1. Linear Least Squares
2. Clustering Classifier
3. Conditional Random Field

We evaluate the proposed models in Chapter 6 for the three sport disciplines mountain biking, cycling, and running.

5.1 Data Pre-processing

The models we show in Section 5.2 require some constraints to be fulfilled by the input data. We performed pre-processing steps to bring the input into normalized and consistent form. Additionally, some feature extraction methods were established to improve the overall model.

5.1.1 Timing

In the individual collection data set, the different features have different update frequencies and therefore there is no regular time interval in which a value for each feature is available. For use in model generation however, values of all features at the same time are needed.

We define a regular interval $t_{interval} = 5\text{ s}$, as the temporal distance between two sets of feature values. The feature values $\hat{F}^i(t)$ at time t for feature i are calculated as the average of all discrete, available raw values $F^i(tt)$ with $tt \in (t - t_{interval}, t]$.

The interval was chosen to be 5 s so it is very likely, that for each feature at least one data point exists within this interval. The GPS location and therefore the speed v has the lowest update frequency and usually has a value every 2 s to 4 s, which makes 5 s a suitable while conservative fit.

5.1.2 Delay

When the intensity of a workout changes, the heart rate adapts to match the new energy consumption. This does however not happen immediately because the chemical processes in the body take place with a feedback delay of a few seconds. Figure 5.1 shows a short extract of a running session where we can see, that the heart rate reaches its maximum point (at $t \approx 450\text{ s}$) about 30 s after the maximum grade has been reached (at $t \approx 425\text{ s}$). For understandability, the elevation is shown as well, while the practically constant cadence and speed are omitted.

This delay varies from subject to subject. Especially the heart rate recovery time after increased effort has been suggested to be a viable measure for fitness as the heart rate returns back to a normal level quicker for fitter people. [49]

The implication of this delay is, that a heart rate model should not just use the current feature values to model the current heart rate but rather take feature values from the last few seconds into consideration. One way to achieve this, is to increase the order of a model to respect multiple previous values.

For our linear least squares model (see Section 5.2.1), we averaged all values over two time frames to produce a representation of not only the current feature value but also the development of the feature over the last few seconds. In contrast to the averaging described in Section 5.1.1, the window size is with a size of 20 s larger and both windows are shifted from each other. The first window (distant window) sums up values from 40 s to 20 s ago, while the second one (close window) includes values from 20 s to 0 s ago.

Figure 5.2 shows the raw vertical speed in blue and both window values for a small extract of a jogging session (red and yellow). For better understandability,

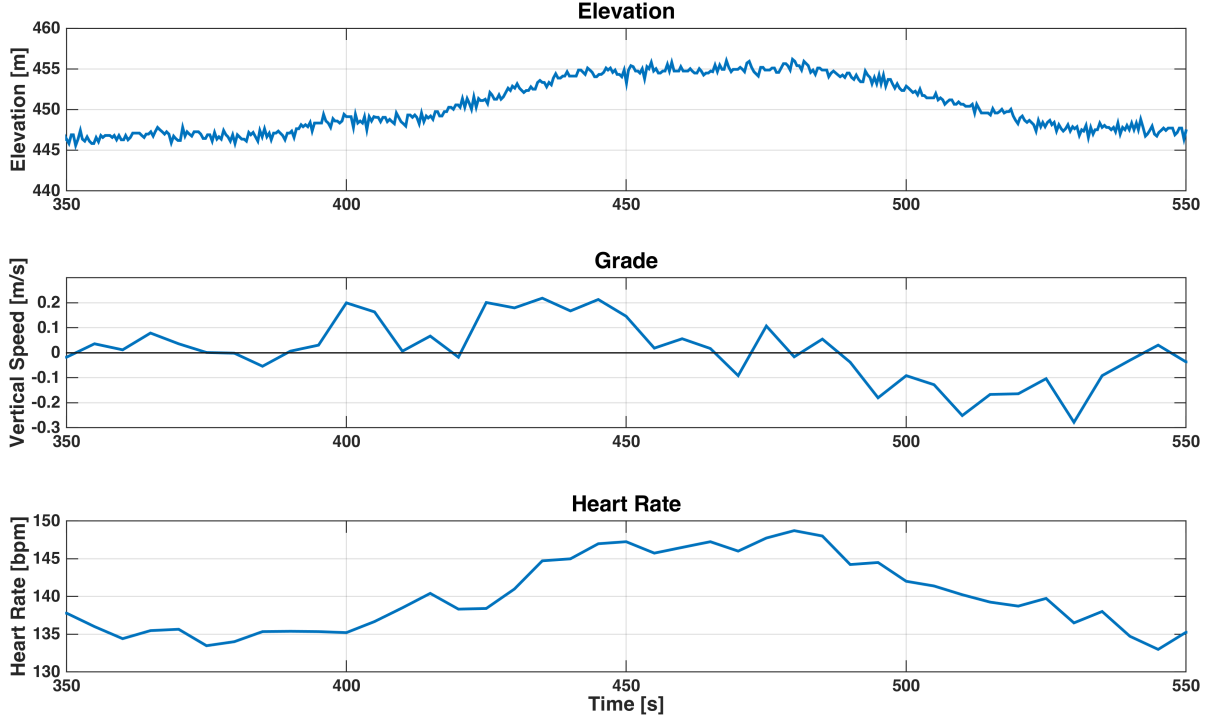


Figure 5.1: Elevation, grade, and heart rate in a short 200 s extract of a workout session. The grade reaches its maximum value around $t \approx 425$ s, while the heart adapts with some delay to the increased intensity and reaches the maximum value about 25 s later at $t \approx 450$ s. The speed and cadence during this workout extract are essentially constant and not shown here.

the plot shows the averaged rather than the summed up values (sum is divided by 4 since it is 4 values each 5 s summed up).

We chose to use two separate windows to enable the model to adapt to different delays for different users. For example, the model could weigh one window more than the other to account for a delay between both windows. The window sizes were chosen based on observations in multiple data sets similar to the extract shown in Figure 5.1, where the delay is around 25 s.

5.1.3 Noise

The raw feature data is usually distorted by some noise. However, the effect of modeling the feedback delay as the sum over multiple values, has an effect of smoothing out the data. There are no further de-noising actions involved at this pre-processing step.

As seen in Section 4.1.1, some measures to minimize noise are already taken

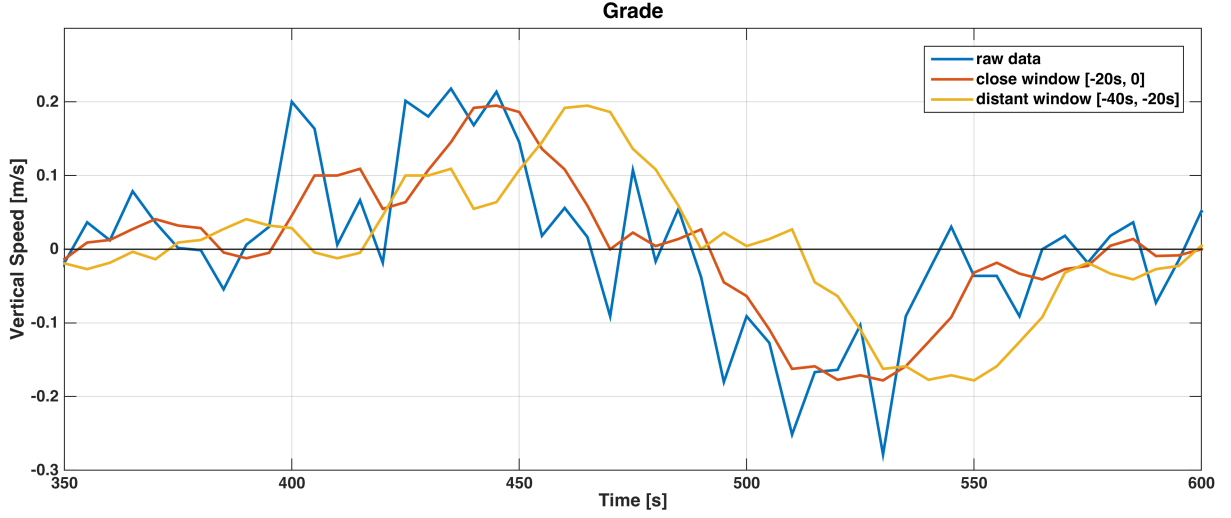


Figure 5.2: Example vertical speed over time during workout session showing raw feature and both averaged windows. We can also identify the smoothing effect the averaging entails. The delayed features are roughly shifted by 10 (close window) and 30 seconds (distant window) to the raw data.

in the data collection phase. For example the instantaneous speed is calculated from the distance between multiple data points rather than a single pair.

5.1.4 Intensity

An alternative approach to feeding many features into a heart rate model is to deduct a more general and expressive measure from multiple raw features. Such a measure could be representing the current intensity of a workout session.

We investigated the possibility to combine speed and grade into a measure of intensity that is directly related to the energy consumption during a workout. Based on the studies by Margaria et al. [8] from 1963 and Minetti et al. from 2002 [10] about the energy cost of running uphill and downhill slopes, we established an intensity factor, which models the energy used for different slopes.

Figure 5.3 shows the energy curve for varying grade. It is based on the metabolic energy cost of running in [10] which is given in $\text{J kg}^{-1} \text{m}^{-1}$ and then normalized to be 1 on flat terrain.

Equation (5.1) states the intensity factor function $F(g)$ for a given grade $g \in \{g \in \mathbb{R} \mid -0.45 \leq g \leq 0.45\}$.

$$F(g) = 43.17g^5 - 8.44g^4 - 12.03g^3 + 12.86g^2 + 5.42g + 1 \quad (5.1)$$

The intensity factor is a more accurate representation of the actual energy

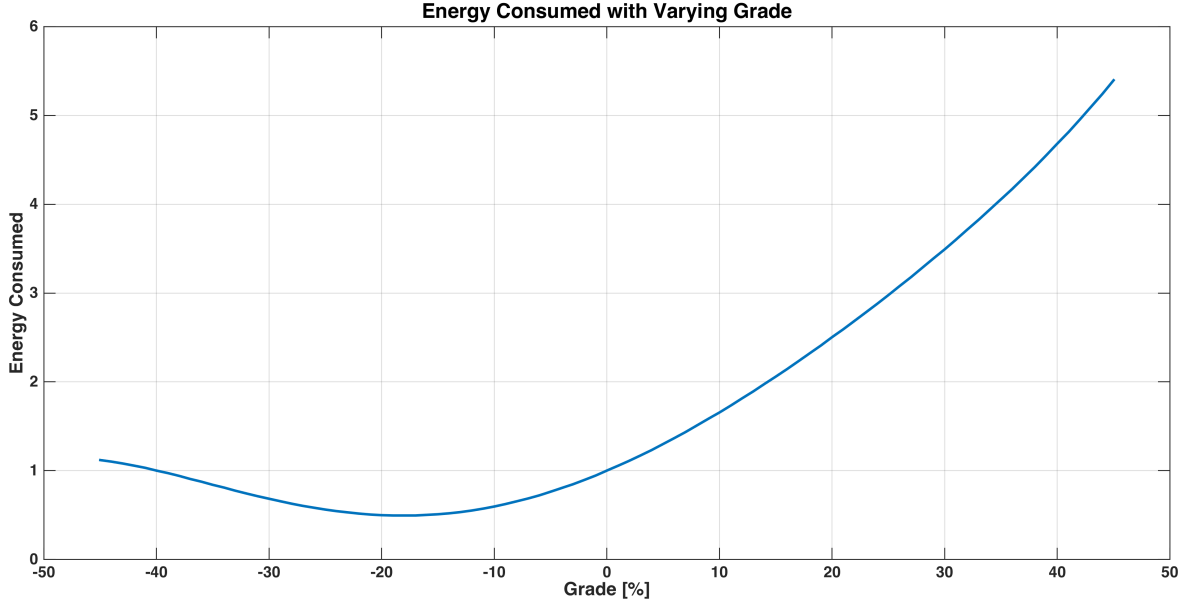


Figure 5.3: Factor of energy consumed while running at constant speed with varying grade. Normalized to flat surface being factor 1. Interestingly, the energy consumption begins to rise again with increasing downhill slopes ($< -18\%$). [10]

consumed during a workout session. When assuming a linear relation of the energy consumption with the speed as well, we can multiply the relative intensity factor $F(g)$ derived from the grade with the speed v to receive our new intensity feature $I(v, g)$:

$$I(v, g) = v \cdot F(g) \quad (5.2)$$

The proposed intensity measure is however not part of any heart rate model we present. Some tests have shown, that it is prone for fluctuations due to noise. We did therefore not further investigate this measure.

5.2 Model Approaches

In this section we explain the three heart rate model approaches in detail. For an extensive comparison, we refer to Chapter 6.

Our heart rate models are trained using supervised learning. The input data consists of a number of feature values as column vectors which are combined into a single matrix $F = (f_1, \dots, f_n)$ (for n features). The according label vector h contains the measured heart rate values. Each row i in F therefore builds a training example with h_i as a label. When training a model function $\hat{h} = M(F)$

we minimize the empirical risk by seeking the function $M(F)$ that best fits the training data and minimizes the norm of the error $\|r\|$ with:

$$r = M(F) - h = \hat{h} - h \quad (5.3)$$

We can then apply M on a test set to retrieve an estimate heart rate \hat{h} and compare it to the measured values h in order to improve our models further.

5.2.1 Linear Least Squares

We introduce a simple linear least squares model based on the assumption that the measured features have a linear relation to the heart rate during an exercise session.

During aerobic exercise where the consumption of oxygen adequately meets energy demands, a linear relation has been suggested in literature for speed [11], even with varying grade [8]. [50] suggests, that even during acceleration and deceleration cycles while running, the energy cost does not significantly vary in relation to the acceleration but rather directly to the speed adopted.

The grade however does not follow a simple first order relation, as seen in Figure 5.3. For our least squares approach we therefore model the grade as second order relation with a quadratic and a linear part.

To account for the cardiovascular drift, we add the time as a feature for our model. It increases steadily over workout duration, much like we expect the heart rate to rise over workout duration due to the cardiovascular drift.

We also add a constant to incorporate the resting heart rate into the model, ending up with the following features:

1. Speed
2. Cadence
3. Grade
4. Time
5. Constant

Formally, we define a matrix $M \in \mathbb{R}^{n \times 6}$ containing the n values per input feature as column vectors ($\in \mathbb{R}^n$):

$$M = \begin{pmatrix} | & | & | & | & | & | \\ t & c & v & g & g^2 & 1 \\ | & | & | & | & | & | \end{pmatrix} \quad (5.4)$$

with:

t : time since start of workout [s]
 c : cadence [s^{-1}]
 v : speed [m s^{-1}]
 g : vertical speed (grade) [m s^{-1}]
 h : heart rate [s^{-1}]

Note, that the last column vector is filled with constant 1, to represent the constant factor in our model. We want to find a vector $x \in \mathbb{R}^6$ satisfying

$$\|Mx - h\|^2 = \min_{\tilde{x} \in \mathbb{R}^6} \|M\tilde{x} - h\|^2 \quad (5.5)$$

With the least squares method, we minimize the residual $r = h - Mx$ to find the unique optimal solution for x , resulting in factors of a simple equation to estimate the heart rate \hat{h} at time t , based on current feature values. The heart rate model function $\hat{h}(t)$ is shown in Equation (5.6).

$$\hat{h}(t) = x_1 \cdot t + x_2 \cdot c(t) + x_3 \cdot v(t) + x_4 \cdot g(t) + x_5 \cdot g(t)^2 + x_6 \quad (5.6)$$

Adding Delay

The model function presented in Equation (5.6), does not yet consider the delay in heart rate adaption to changed workout intensity. We apply the method introduced in Section 5.1.2 to receive two representations of each feature as sums over separate time frames of 20 s length. We recall, that the distant window sums up values from 40 s to 20 s ago, while the close window considers the last 20 s.

We replace the matrix M with a version $M_d \in \mathbb{R}^{n \times 10}$ containing the two sums columns for the features cadence, speed and grade (as well as the squared grade). Therefore we have $4 \times 2 + 2 = 10$ free variables in x now. The least squares approach can equivalently be applied.

Multiple Workouts

The training process of the linear least squares model cannot only be done for a single workout session. By adding data from multiple workouts to the feature matrix and heart rate vector, we can train the free parameters on a large set of workouts simultaneously. Training based on multiple workouts is expected to yield a more general model that performs better on new workouts. This is because different kinds of workouts are included and the feature value variation can be expected to be bigger, therefore modeling more possible developments during a workout session.

The number of rows in the learning feature matrix F , changes to be $n = n_1 + \dots + n_k$ with n_i being the number of feature tuples in workout i out of k total workouts. It is however essential to note, that we still train the model per user, since we expect significant differences between multiple users.

5.2.2 Conditional Random Field (CRF)

A Conditional Random Field (CRF) is a discriminative, probabilistic model to segment and label sequence data. They were first introduced by Lafferty et al. [51] in 2001 and are used for structured prediction with supervised learning.

Given observations x and labels y , we want to find a model, that can predict y accurately with x as input. In our use case, the input observations are structured as a sequence of feature values during a workout session. The literature describes multiple modelling methods for structured prediction, the most known probably being Hidden Markov Models [52].

Compared to a Hidden Markov Model, a CRF does not model the joint distribution $p(x, y) = p(y|x)p(x)$ but only the conditional distribution $p(x|y)$, which is sufficient for classification. In many application domains, modeling the joint distribution is hard because $p(x)$ contains highly dependent features. By directly modeling the conditional distribution $p(x|y)$, the independence assumption for observations x can be relaxed. We however lose the ability to use the model in a generative way to produce random observation sequences (we can easily do this with markov models). This kind of model is called a discriminative model.

For our application, we concentrate on the linear-chain CRF model, which can be understood as the conditional version of a HMM, as it also has a chain as an underlying structure. This directly reflects the sequential observation data we have from our workouts.

For labels $y \in Y$ and observations $x \in X$ with a parameter vector $\lambda \in \mathbb{R}^k$, the linear-chain Conditional Random Field is defined as a distribution $p(y|x)$ as follows [53, 54]:

$$p(y|x; \lambda) = \frac{1}{Z(x; \lambda)} \exp \left\{ \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, x_t) \right\} \quad (5.7)$$

where $Z(x; \lambda)$ is an instance-specific normalization function

$$Z(x; \lambda) = \sum_y \exp \left\{ \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, x_t) \right\} \quad (5.8)$$

The $k \in \{1, \dots, K\}$ feature functions f_k , are a mapping $f_k : X \times Y \rightarrow \mathbb{R}$. The parameters λ_k are weights for each feature function.

Generally, feature functions can be arbitrary complex. For applications where real-valued observation data should be modeled in a linear-chain CRF, it is however common to use two main kinds of feature functions: [55]

Edge-Observation Feature Functions

Features that only depend on the last state y_{t-1} and current state y_t .

Node-Observation Feature Functions

Features that are only dependent on the current state y_t and the current observation x_t .

The use of these two types of feature functions drastically reduces model complexity and resembles the two main parts of a HMM: the state transition probabilities and the output probabilities.

To define these feature functions formally, we use the notation $1_{\{\text{condition}\}}$ to denote a function having value 1 when the given condition is met and 0 otherwise. The function $q_m(x_t)$, gives the m^{th} observation feature value of the observation $x_t \in X$ where $m \in \{1, \dots, M\}$ stands for the observation features like speed, grade, or cadence. (E.g., $q_2(x_t)$ stands for the speed at x_t)

For our model, we define edge-observation feature functions as seen in Equation (5.9) and node-observation feature functions as shown in Equation (5.10).

$$f_k(y_{t-1}, y_t) = 1_{\{y_{t-1}=y\}} 1_{\{y_t=y'\}} \quad \forall (y, y') \in Y \times Y \quad (5.9)$$

$$f_k(y_t, x_t) = q_m(x_t) \cdot 1_{\{y_t=y\}} \quad \forall y \in Y, \forall m \quad (5.10)$$

Using these two types of feature functions, we end up with $|Y|^2$ edge features and $M \cdot |Y|$ node features, totalling for $K = |Y|^2 + M \cdot |Y|$ feature functions. Each of these feature functions is assigned a weight λ_k during training. For our default configuration, we use $M = 4$ observations (time, speed, grade, and cadence) and $|Y| = 18$ labels (see Section 5.2.2), resulting in $K = 396$ free variables to be trained.

Inference

There are two inference problems to be solved when using a CRF:

1. During training, compute the weights λ
2. For classification, compute the most likely labeling

For the linear-chain CRF, both tasks can be efficiently solved using variants of standard algorithms for HMMs. Commonly, the log likelihood

$$l(\lambda) = \sum_{i=1}^N \log p(y|x; \lambda) \quad (5.11)$$

is maximized. For a linear-chain CRF this function is concave, which means any local maximum is also the global maximum. This allows us to use a gradient-ascent algorithm to find the optimal solution. [55] However, this typically requires too many iterations to be practical. Researchers have suggested the use of quasi-Newton methods such as BFGS [56, 57], which do not need to compute and store entire Hessian matrices (compared to the Newton method). The Matlab CRF library we used comes with an efficient BFGS implementation.¹

The second inference problem, is finding the most likely labeling:

$$y^* = \arg \max_y p(y|x; \lambda) \quad (5.12)$$

Using a slightly modified Viterbi algorithm, this maximum marginal distribution can efficiently be calculated. As described in Section 5.2.2, the output of the classification step is a probability distribution $W \in [0, 1]^{|S| \times n}$ for every possible label $s \in S$ at any observation n .

Input

In the supervised learning phase for our CRF model, we have the following inputs:

1. Sequences of Observations $x \in X$
2. Sequences of according Labels (States) $y \in Y$

We recall, that our goal is to model the heart rate based on several input observations. Therefore, the labels (we call them states here) needed for training are deducted from the heart rate. To reduce the complexity of the model, the state space Y does not include all possible heart rates but rather a projected version to reduce the number of states.

We define a projection function $h : \mathbb{R} \rightarrow S \subset \mathbb{N}$, given a state size δ , as follows:

¹HCRF 2.0 Library
Available at: <http://sourceforge.net/projects/hcrf/> [Accessed 05-13-2015]

Property		Value
Minimum Heart Rate [bpm]	hr_{min}	30
Maximum Heart Rate [bpm]	hr_{max}	200
State Size [bpm]	δ	10
Number of States	$ S $	18

Table 5.1: Default configuration values for the heart rate projection function h .

$$h(y) = \begin{cases} 0 & \text{if } y \leq hr_{min} \\ \text{round}((y - hr_{min})/\delta) + 1 & \text{if } hr_{min} < y < hr_{max} \\ \lceil (hr_{max} - hr_{min})/\delta \rceil & \text{if } y \geq hr_{max} \end{cases} \quad (5.13)$$

Therefore, every label $h(y)$ represents $\delta = 10$ possible whole numbered heart rates in the range $[hr_{min}, hr_{max}]$. We can now define the state space S more precisely as:

$$S = \{s \in \mathbb{N} | 0 \leq s \leq \lceil (hr_{max} - hr_{min})/\delta \rceil\} \quad (5.14)$$

Based on literature on maximum heart rate [21, 32, 33] and own observations, we chose hr_{max} and hr_{min} as seen in Table 5.1. In Section 5.2.2, we discuss how we can represent heart rates between two states and therefore reach a higher resolution than the selected $\delta = 10$.

Output Interpretation

When using the linear-chain CRF for prediction of the heart rate, given sequences of observations, we obtain an output matrix $W \in [0, 1]^{|S| \times n}$ per test sequence, containing probabilities for every state $s \in S$ at observation n .

To convert those probabilities back to a heart rate, we consider two approaches for $i = 1, \dots, n$:

1. Select the state s with highest probability

$$\eta_i^r = \arg \max_{s \in S} W_{s,i}$$

2. Calculate the weighted average of the states S (as a column vector) using the probabilities $W_{\cdot,i}$, yielding inter-state values.

$$\eta_i^w = S \cdot W_{\cdot,i}$$

By applying the inverse h^{-1} of the heart rate to state label projection function (Equation (5.13)), we get the heart rate estimate \widehat{hr} predicted by our CRF model:

$$\widehat{hr} = h^{-1}(\eta) = \eta \cdot \delta + hr_{min} \quad (5.15)$$

Depending on the state size δ , the resolution of the deducted heart rate can be quite low when using the first approach. This is also the reason, why the second method is superior, as it can represent heart rates between multiple states. The figures in Section 6.1.2, show both measures in comparison.

5.2.3 Other Approaches

Apart from the CRF and the linear least squares model, we also tested the performance of a Naive Bayes Classifier. It does however lack the property to embed the sequential structure of our data. This was clearly visible in the prediction quality and we quickly went to a more powerful and sophisticated approach using the CRF model.

5.3 Fitness

In Section 2.4 we discussed how $VO_2 \max$ can be used as a fitness measure. However, it is not feasible to determine measures like HR_{\max} or HR_{rest} during normal workout sessions. We propose another fitness measure, that is not directly coupled to a $VO_2 \max$ measurement or approximation. Another direct consequence of improved fitness is a lower heart rate for equal workout intensity. This is not only because well trained athletes need less energy [8], but also because the heart becomes stronger and can more efficiently move O_2 through the body with every heart beat. This is also directly reflected by a lower resting heart rate HR_{rest} . [31]

We aim at using the information from a trained linear-chain CRF model, to tell a subject when his or her fitness improves. The weights λ_k incorporate valuable information about state transitions and the size of observation values in a given state. The proposed fitness measure is motivated by the assumption, that the fitness level of a subject gradually increases with regular exercise. By training the CRF model for different subsets of a subject's workouts, we are able to investigate the changes in the trained model weights.

5.3.1 Weight Properties

In Figure 5.4, we show a visualization of the edge-feature weights λ_e and the node-features weights λ_n after training the CRF heart rate model on the first

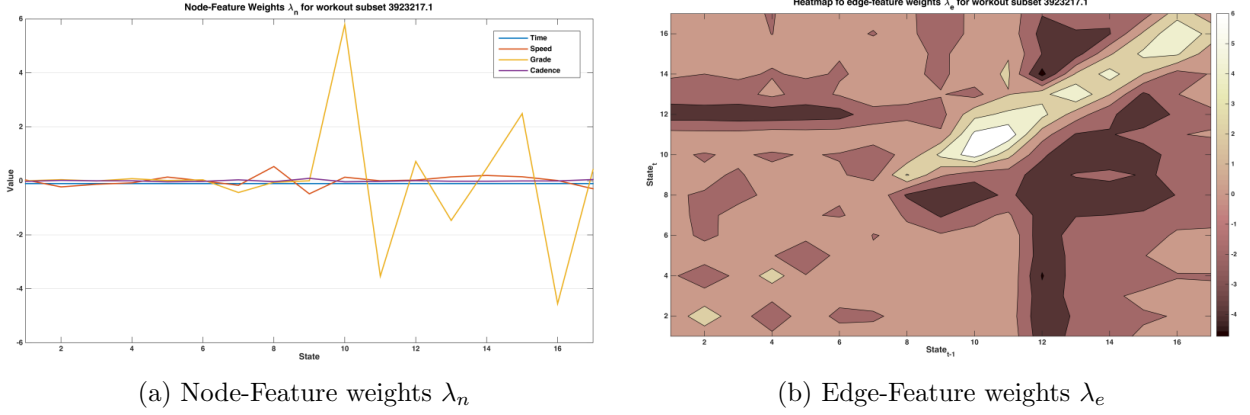


Figure 5.4: Visualization of the trained model weights λ for the first 10 workouts of user 3923217. The node-feature weights λ_n reflect the amount of codetermination each observation feature has on the predicted heart rate. The edge-feature weights λ_s determine the probability of a state transition.

10 workouts of subject 3923217. As a comparison, Figure 5.5, shows the same visualization for the 8th subset of workouts for the same subject, corresponding to workout 71 to 80 in our collection for said subject. We assume, that the subject had a better fitness level during the workouts in the second data subset (as there were at least 60 workouts between the two subsets).

Some interesting observations include the lower magnitude of positive node-feature weights in the second set and the drift of high edge-feature weights to lower states. For the edge-feature weights, we are mainly interested in the diagonal band from the bottom left to the top right, where the weights represent the probability of the next state remaining unchanged or shifting by exactly one. We do not expect the actual heart rate to shift heavily from one observation to the next. The model accurately reflects this, as all the weights around the diagonal are comparably low.

To learn how the model weights have an influence on the fitness level of a subject, we define a set of weight properties that can be calculated for each trained model based on its learned weights λ . We then analyze which of these properties increase and decrease for subsequent workout sets $m + 1$ and $m + 2$ of the same subject. Each subset consists of 10 workout sessions based on which the CRF model is trained. Weight properties that consistently take the same direction in most comparisons, are then proposed for use in the fitness model.

Table 5.2 shows the percentage of weight comparisons for subsequent subsets that increased and decreased in value. In the table, we make use of the variable b_e , defined as a vector of all the edge-feature weights λ_e on the main, the 1st and the -1st diagonals. We also define a function $c(S; \lambda_s) : \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{R}$ to represent the weighted center state of the state-feature weight vector $\lambda_{s,i}$ of observation

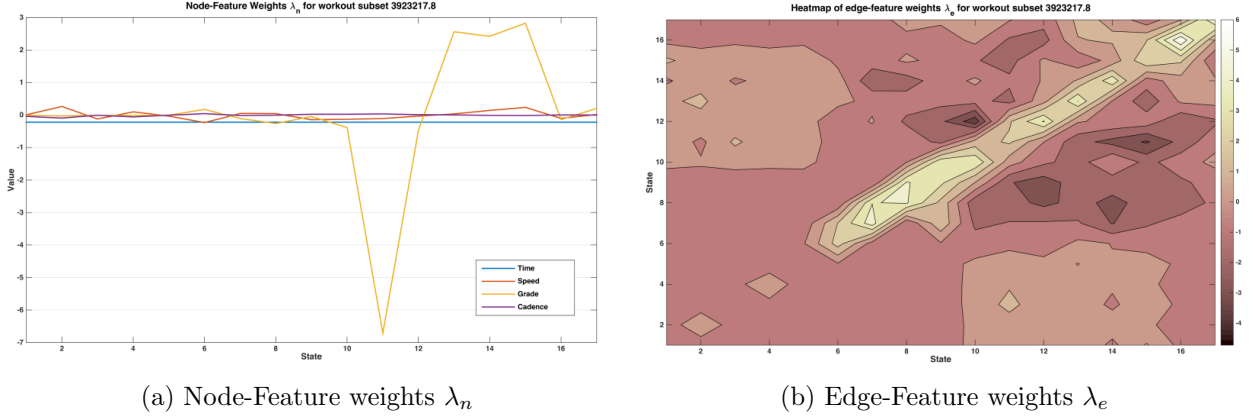


Figure 5.5: Equivalent visualization to Figure 5.4 for workouts 71 to 80 in the test set of subject 3923217. Note that the highest edge-feature weights are in lower states than before and the magnitude of the positive node-feature weights is lower.

feature i as follows:

$$c(S; \lambda_{s,i}) = \langle S, \frac{\lambda_{s,i}}{\sum_t \lambda_{s,i}(t)} \rangle \quad (5.16)$$

with $S = (1, \dots, |Y|)$

We recall, that Y is defined as the set of possible integer states the output labels of the CRF model can adopt.

We propose a classification function improvement $(\lambda^{old}, \lambda^{new}) : \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{B}$, that takes a previous and a new model weight vector as an input and returns a boolean **True** if it is confident that the fitness level has improved between the two workout sets that were used to train the input weights.

The following list, briefly describes some of the properties and motivates their use as part of a fitness improvement function:

Decreasing $\text{mean}(\lambda_{s,3})$ (grade)

As $\lambda_{s,3}$ directly represents the impact of the grade on the predicted heart rate, a decline of its values can be interpreted as the grade taking a smaller part of the intensity. This is reasonable as the energy consumption on slopes has been shown to be 5 % to 7 % lower for trained athletes compared to untrained subjects. [8]

Decreasing $\|\lambda\|$

A decreasing norm of the overall weight vector implies, that the overall

Property	Model $m - 1$		Model $m - 2$	
	fall	rise	fall	rise
# States	21.88 %	21.88 %	23.08 %	15.38 %
$\max \lambda_e$	43.75 %	56.25 %	69.23 %	30.77 %
$\max \lambda_s$	46.88 %	53.13 %	46.15 %	53.85 %
$\text{mean}(\lambda_e)$	43.75 %	56.25 %	53.85 %	46.15 %
$\text{mean}(\lambda_s)$	56.25 %	43.75 %	46.15 %	53.85 %
$\text{std}(\lambda_e)$	43.75 %	56.25 %	69.23 %	30.77 %
$\text{std}(\lambda_s)$	50.00 %	50.00 %	46.15 %	53.85 %
$\text{mean}(b_e)$ (Edge Band)	43.75 %	56.25 %	76.92 %	23.08 %
$\text{mean}(\lambda_s^2 \lambda_s > 0)$	43.75 %	56.25 %	46.15 %	53.85 %
$\text{mean}(\lambda_s^2 \lambda_s < 0)$	53.13 %	46.88 %	61.54 %	38.46 %
$\text{mean}(\lambda_{s,1})$ (time)	50.00 %	50.00 %	38.46 %	61.54 %
$\text{mean}(\lambda_{s,2})$ (speed)	50.00 %	50.00 %	53.85 %	46.15 %
$\text{mean}(\lambda_{s,3})$ (grade)	59.38 %	40.63 %	76.92 %	23.08 %
$\text{mean}(\lambda_{s,4})$ (cadence)	46.88 %	53.13 %	53.85 %	46.15 %
$c(\lambda_{s,2})$ (speed)	62.50 %	37.50 %	46.15 %	53.85 %
$c(\lambda_{s,3})$ (grade)	56.25 %	43.75 %	61.54 %	38.46 %
$c(\lambda_{s,4})$ (cadence)	43.75 %	56.25 %	38.46 %	61.54 %
$\ \lambda\ $	46.88 %	53.13 %	69.23 %	30.77 %
$\ \lambda_e\ $	43.75 %	56.25 %	53.85 %	46.15 %
$\ \lambda_s\ $	43.75 %	56.25 %	69.23 %	30.77 %

Table 5.2: Percentage of cases, where the listed property for a trained CRF model m increased or decreased compared to model $m - 1$ or $m - 2$, corresponding to the last and second to last subset of 10 workouts of the same subject, respectively. Weight properties that rise or fall in more than 60 % of our tests are colored in red.

heart rate will be lower for similar input observations and hence similar intensity. This is another reasonable property of increased fitness.

Decreasing $\|\lambda_s\|$

As the state-feature weights represent the impact of different observation features on the predicted heart rate, decreasing values lead to a generally lower prediction. Again, a change we would expect with increased fitness.

Decreasing $\text{mean}(b_e)$

Closely related to the overall norm decreasing, declining values in the most important transition weights lead to a lower heart rate if the effect is not compensated with higher state weights λ_s . As this is not the case, this property appears to be a suitable component of a fitness measure as well.

Increasing $\text{mean}(\lambda_{s,4})$ (cadence)

We assume, that two factors play a role in the increase of the cadence impact on the predicted heart rate: (a) The weight increases to compensate a slightly decreased cadence for fitter subjects with the same heart rate; and (b) with improving fitness, the stride technique improves as well and becomes more efficient. The impact compared to other observation features therefore increases. For these effects to reflect in the model weights, we however expect more time to be needed than with other property changes that also showed higher occurrence rates.

Based on these observations, we propose to use the properties with a change in more than 65 % of our tests to build our improvement function. These reflect the properties that we can motivate to have a plausible impact on the overall fitness of a subject. We describe the exact functionality of the improvement function in Section 6.2.

Results

In this chapter, we compare how the proposed models perform for running workout sessions. In a first part, we discuss the linear least squares and CRF heart rate models and show how accurate their heart rate predictions are. Various tests are performed to optimize model parameters and compare the model performance under different conditions and for different sports.

In a second part, we show results for the proposed fitness model.

6.1 Heart Rate

The main aspect of this thesis is the testing and evaluation of different heart rate models. In the following sections, we present testing and evaluation results for the previously introduced Linear Least Squares Conditional Random Field heart rate models.

6.1.1 Linear Least Squares

As shown in Section 5.2.1, the input for the linear least squares model are a matrix $M \in \mathbb{R}^{n \times 6}$, containing the observation data, and a vector $h \in \mathbb{R}^n$ containing the according heart rate data for training.

In the following sections, we show the heart rate as modeled by the linear least squares model in comparison to the actually measured heart rate. All data used is from the individual collection data set (see Section 4.1).

For testing purposes, we usually trained the model on one workout and measured the error of the model prediction for the same workout. This gives a good indication of what the best prediction we could expect looks like, when later applied generally for models over multiple workouts.

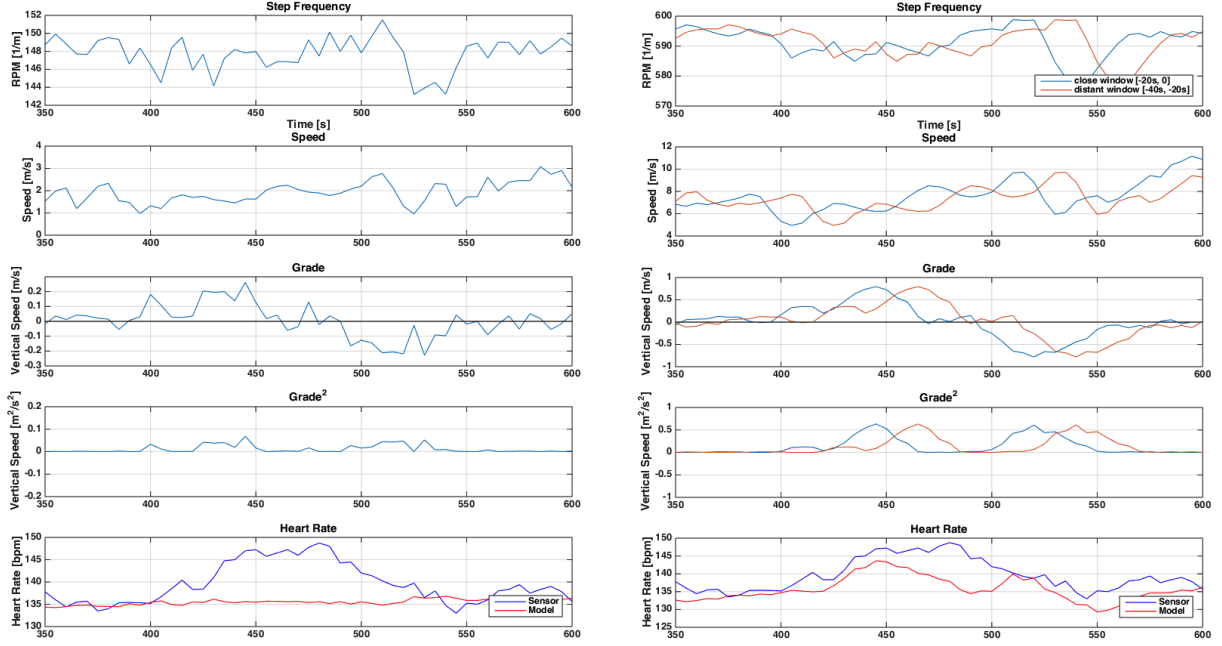


Figure 6.1: Comparison of model prediction for a small hill with delay (right) and without delay (left). Without the delay modeled, the rising heart rate during the ascent can not be modeled.

Delay Impact

As discussed in Section 5.1.2, we model the heart rate adaption delay using two windows over which observation values are summed up. In Figure 6.1, the same short hill is shown with corresponding observation data. The heart rate plots include both the measured heart rate from the chest strap and the predicted heart rate from the model, which was trained on the data from the shown workout itself.

The complete heart rate prediction for the same workout, is shown in Figure 6.3. The error norm for this prediction is 2.9 bpm with an standard deviation of 2.3. Many smaller heart rate deviations are also recognizable in the prediction heart rate, while some developments are not correctly reflected by the model. Such instances can be seen at around 1100s where the subject had to stop at a red light

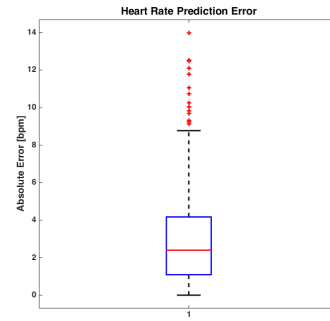


Figure 6.2: Boxplot of absolute heart rate prediction errors for the linear least squares model trained on one workout also used for prediction.

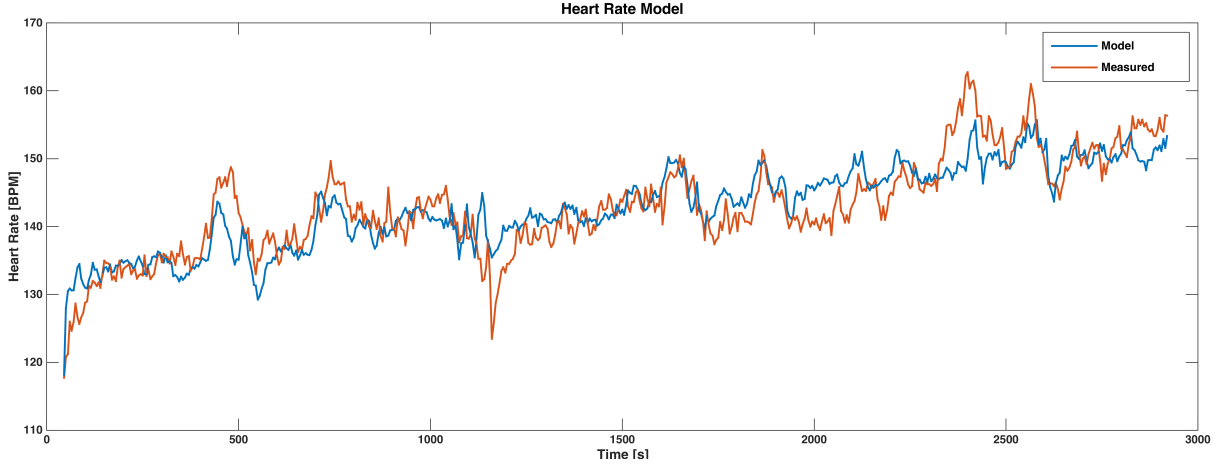


Figure 6.3: Example comparison of model prediction to measured heart rate for a workout of about 50 min length. The prediction is reasonably accurate, but there remain some parts where the model is unable to reflect a heart rate change. (see text)

on a crosswalk and around 2400 s. The exact reason for this peak in heart rate is unknown to the writer.

Figure 6.2, shows the corresponding error distribution as a boxplot. We note, that there are many outliers, representing the previously mentioned parts of the workout, where the model was unable to reflect certain heart rate changes.

Cardiovascular Drift

In Section 3.2, we motivated the need to take the cardiovascular drift into account when building our heart rate model. We recall, that the cardiovascular drift is the observation, that the heart rate slowly increases over time despite constant exercise intensity.

In the least squares model, we added the time as an observation feature to include this effect. For the same workout as shown in the last section, Figure 6.4 shows the achieved heart rate prediction when the time is not included in the observation data.

In the direct comparison, the prediction has a significantly higher error than when the time is included in the model. On the other hand however, the underlying linear function for the time, ensures that the prediction continuously increases. The consequence could be, that this term grows very large for very long workouts, making an accurate prediction practically impossible. Additionally, it cannot account for more complex events like short breaks that usually “restart” the effect of the cardiovascular drift. However, we are confident, that the gen-



Figure 6.4: Heart rate reference and prediction using the linear least squares model without the time as an observation feature. The fit is clearly inferior compared to the one when the time is included. (see Figure 6.3) Note, that the learned parameters for the grade are negative, as the predicted heart rate makes a bow in the wrong direction in places where the heart rate rises due to an ascent.

eral prediction for workouts that are always approximately of the same duration, is significantly better when the time is included. For a better handling of the described shortcomings, a more complex model is needed (see Section 6.1.2).

Multiple Workouts

To consider multiple workouts at once for model training, we extend our definition of the observation matrix M to not only contain the observation vectors for a single workout, but rather all workouts we want to use for model training. More formally, for every workout $i \in \{1, \dots, n\}$, we use their input matrices M_i to build a new matrix M^* as follows:

$$M^* = \begin{pmatrix} M_1 \\ \vdots \\ M_n \end{pmatrix} \quad (6.1)$$

When the model is trained using two workouts, the absolute prediction error for those workouts themselves is in the range 6.5 ± 5.1 bpm as seen in Figure 6.5. When the trained model is used to predict the heart rate for another test workout, the prediction is too inaccurate to be usable. This is primarily because the number of training workouts is too low, but it also suggests that the chosen linear

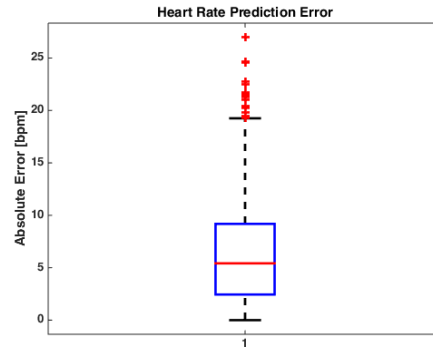


Figure 6.5: Boxplot of absolute heart rate prediction errors for the linear least squares model trained on two workouts also used for prediction.

least squares heart rate model might be too simple for an accurate prediction.

Property	Value
Data Set	
Training Set Size	15
Test Set Size	10
Discipline	running
Observations	Time, Speed, Grade, Cadence
CRF Configuration	
Model Order	1 st
Inference Algorithm	BFGS
Max Iterations	300
Heart Rate Projection Function h	
Minimum Heart Rate	30 bpm
Maximum Heart Rate	200 bpm
State Size	10 bpm

Table 6.1: Default configuration values for the CRF heart rate prediction model.

6.1.2 CRF

In this section, we describe various experiments we conducted to test and improve the CRF model for heart rate prediction.

All experiments were run with the same basic parameters, save the ones being experimented on. In Appendix B.1, we give an overview of the hardware used to run the experiments.

We recall, that the all experiments are conducted on the Endomondo online data test set, if not noted otherwise. Per user in the data set, we split the available workouts in subsets, each containing enough workouts for training and testing (based on the configuration; default is $15 + 10 = 25$). Users with too few workouts are ignored for these tests. The model was then trained on a subset of each user’s workouts and the errors were calculated on the prediction for the according test sets.

Error Measures

We generally measure the prediction error E of the model prediction for the heart rate \widehat{hr} as:

$$E = hr - \widehat{hr} \quad (6.2)$$

Therefore giving us the absolute prediction error. When using the model on multiple workouts (sequences) simultaneously, we consider the mean error of E for every sequence. As shown in Section 5.2.2, we distinguish two ways

to obtain the output state used to calculate the heart rate (using the inverse projection function h^{-1}): (a) The raw state with maximum probability η_i^r for every observation i ; and (b) the weighted state η_i^w , that is obtained by weighing states with their probabilities. For both these measures, we calculate the absolute error E_a and relative error E_r :

$$E_a = hr - h^{-1}(\eta) \quad (6.3)$$

$$E_r = \frac{hr - h^{-1}(\eta)}{hr} \quad (6.4)$$

In the following sections, we present boxplots¹ of the mean errors over multiple test workouts from different users. The first experiment in Section 6.1.2, where we vary the training set size, includes boxplots for all four measures we discussed. Subsequent experiment results will only include the absolute and relative errors for the weighted predictions $h^{-1}(\eta^w)$, as those show strictly better predictions. For reference, we include the other boxplots in Appendix B.2.

We note, that the shown boxplots contain a relatively high amount of outliers with high errors. This is due to the fact, that the Endomondo data has a very diverse set of workouts. Some are of exceptionally good quality, while others lack partial data or have high variations in some observation features. In the end, we have to keep in mind, that all the data was originally gathered on a user's smart phone where Endomondo is out of control of external influences and sensor hardware accuracy.

Training Size

Figure 6.6, shows the relative errors of the heart rate prediction with varying training set size. By directly comparing the adjacent figures, we observe, that the weighted heart rate prediction outperforms the raw heart rate prediction slightly. For the next experiments, we therefore only show the relative and absolute weighted heart rate prediction errors.

For training sizes between 10 and 20 workouts, we observe very similar results with a median weighted heart rate error of about 5.5 %. This corresponds to absolute errors of about 8 bpm as seen in Figure 6.7.

A training size of only 5 workouts seems too low as the 75th percentile of the error is significantly higher. We might expect the error to decrease with more training sequences, but the boxplots show higher errors for training set sizes of 25 and 30 workouts. There are two reasons for this effect:

¹See Appendix A.1 for an explanation about boxplots.

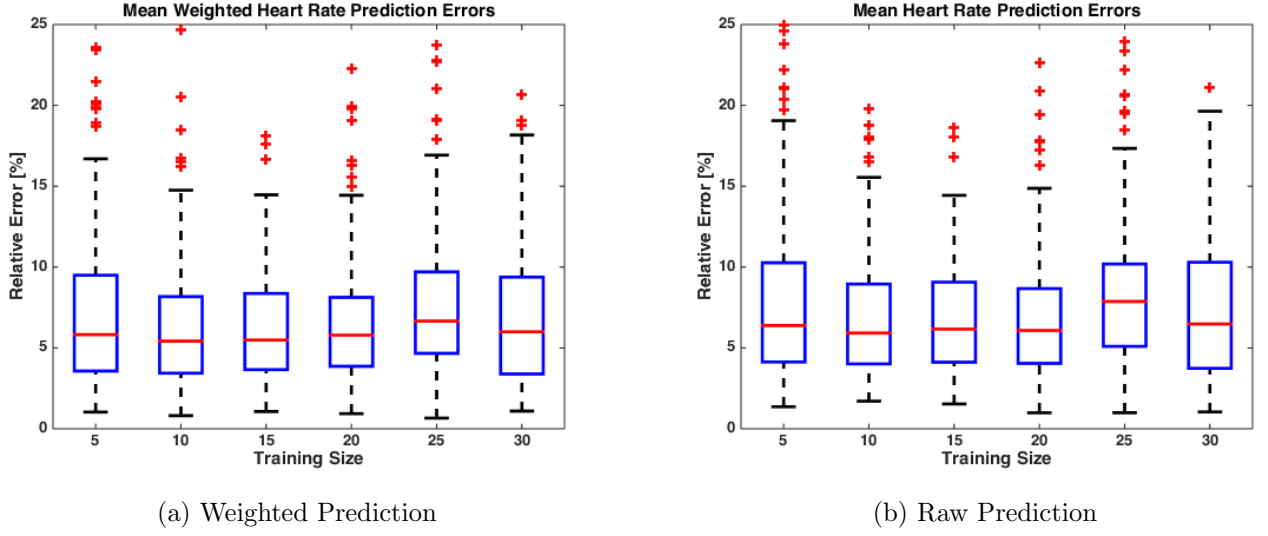
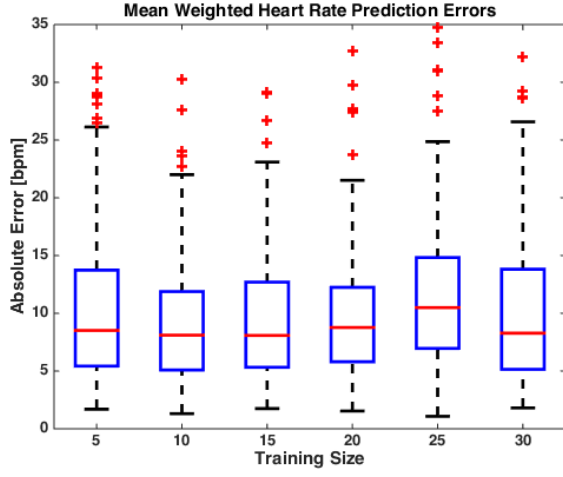


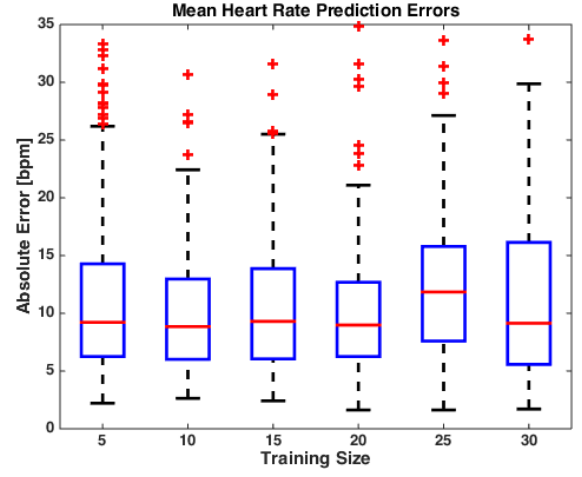
Figure 6.6: Relative errors of the heart rate prediction when varying the size of the training set. We see good values for 10 to 20 workouts for training, while larger training sets show slightly worse results. In all cases, the weighted heart rate prediction measure performs slightly better than the raw version.

1. With increasing training set size, the number of users that have enough workouts decrease. It is therefore possible, that more workouts with inferior data quality and worse fit for the model remain.
2. We expect subjects to improve their fitness over time (see Section 6.2). Since the model reflects the user fitness to some extent, the model trained on many workouts potentially spans over different fitness levels and therefore represents some intermediate fitness level, while the workouts used for testing represent the highest fitness level for the subject up to this point.

Based on this error data, we chose 15 workouts to be a good training set size. A visualization of one heart rate prediction can be seen in Figure 6.8. We note, that the weighted prediction closely reflects the bigger movements of the actual heart rate while being slightly off by about 10 bpm.



(a) Weighted Prediction



(b) Raw Prediction

Figure 6.7: Absolute errors of the heart rate prediction with varying training set size.

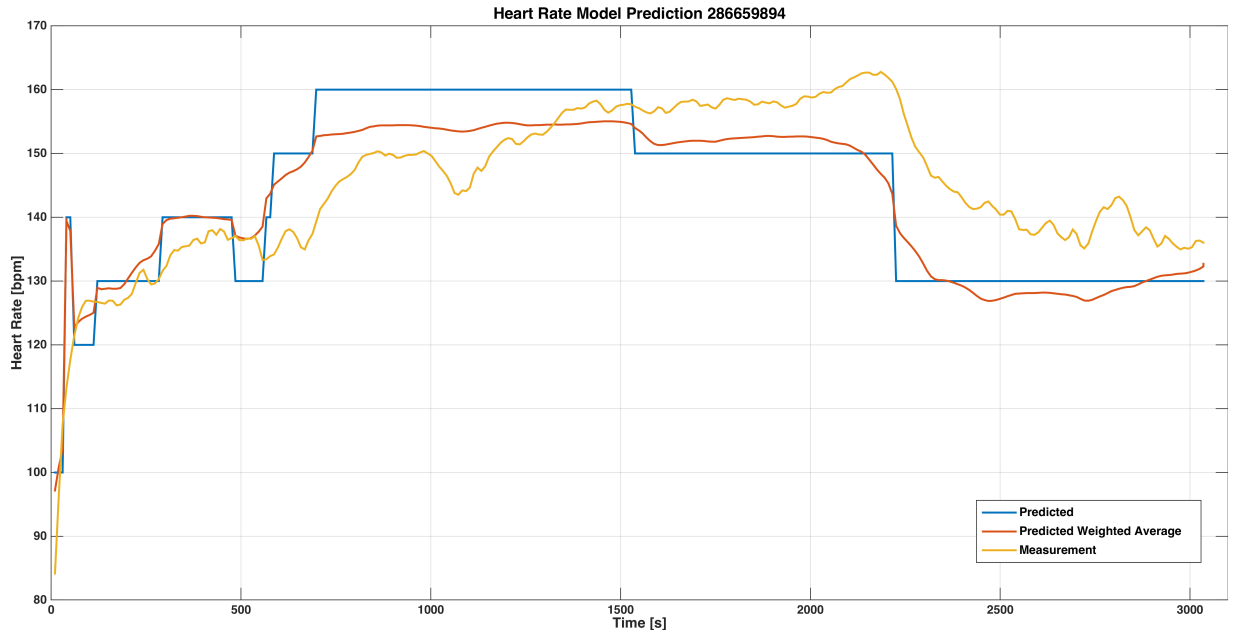


Figure 6.8: Example heart rate prediction for a running workout, showing the true measured heart rate and the raw and weighted predictions.

State Size

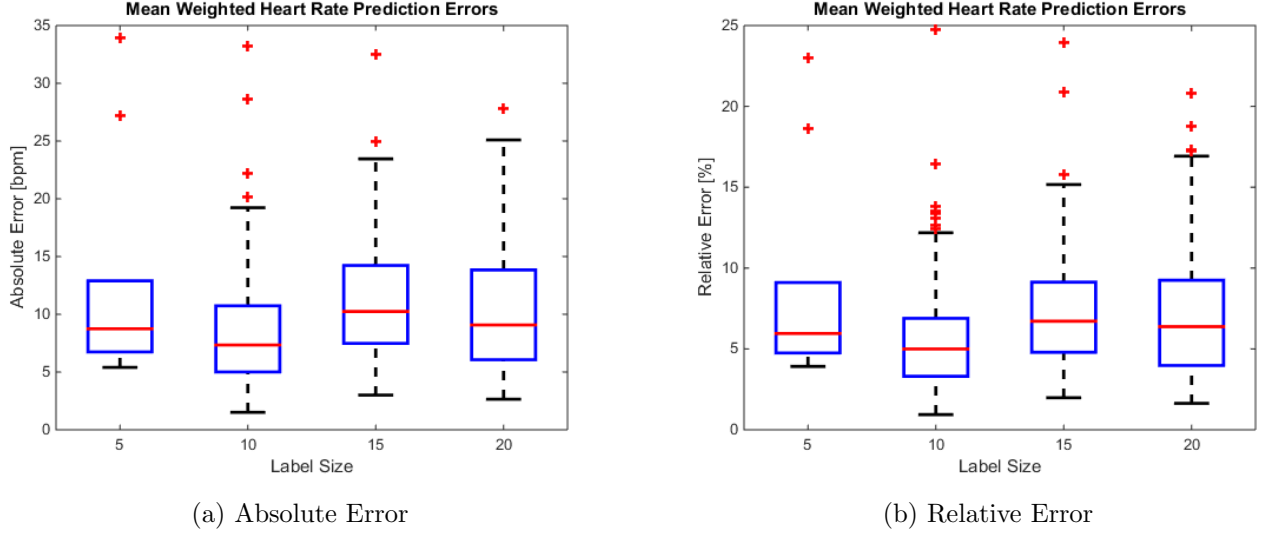


Figure 6.9

Figure B.2 shows the relative and absolute weighted heart rate prediction errors for varying label (state) size δ . The state size is used to project the raw heart rate to a set of labels S as described in Section 5.2.2. The label size directly influences the resolution and maximum accuracy of the prediction. It is unexpected to see, that the error for a label size of 5 bpm is significantly higher than for 10 bpm. We suspect, that the model suffers from overfitting at this point. By halving the label size, the state space doubles and the complexity of the model training process grows quadratically. This is also reflected by the run time of the training process, which was six times higher for $\delta = 5$.

Discipline Comparison

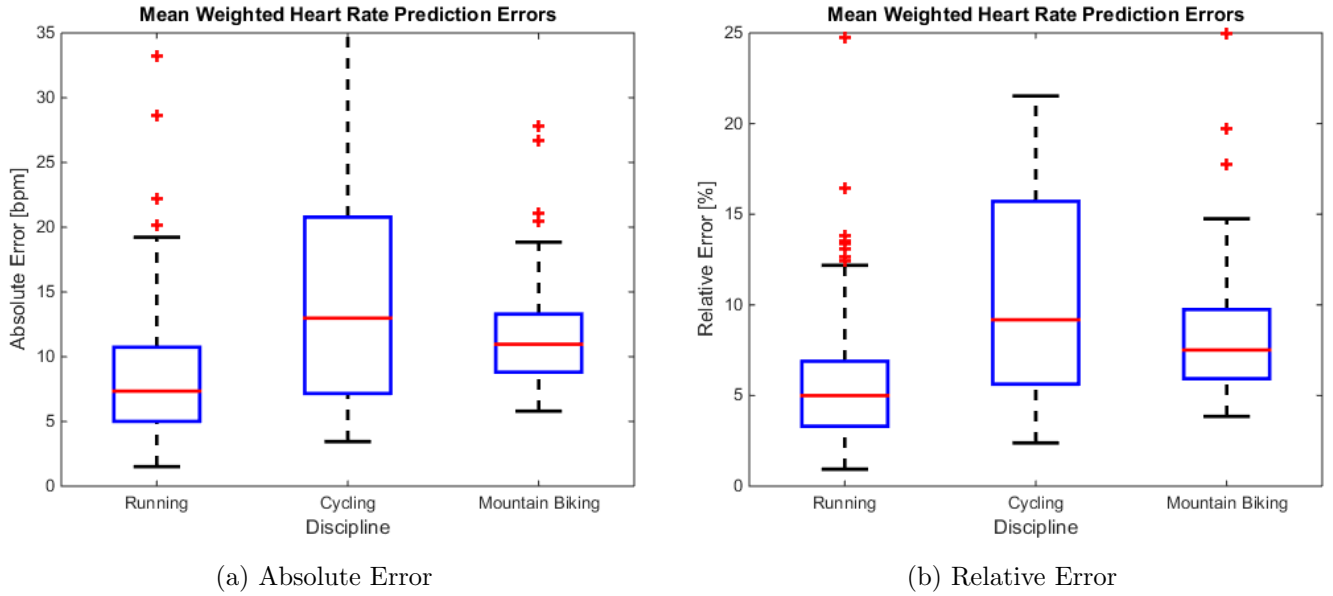


Figure 6.10

Although we focus on running, we also gathered data sets for the disciplines mountain biking and cycling. Figure 6.10 shows the mean errors of heart rate prediction using the model developed for running sessions. While the model generally works on those alternative sports, they still show significantly worse results than for running. One reason that was especially observed for cycling, is the difference in downhill movement between sports on a bike or cycle and running. With a bike, a subject does not have to pedal at all when going downhill, but for sure can. This skews the data sets where the prediction is exceptionally bad in downhill segments as the model fails to incorporate that aspect.

Feature Selection

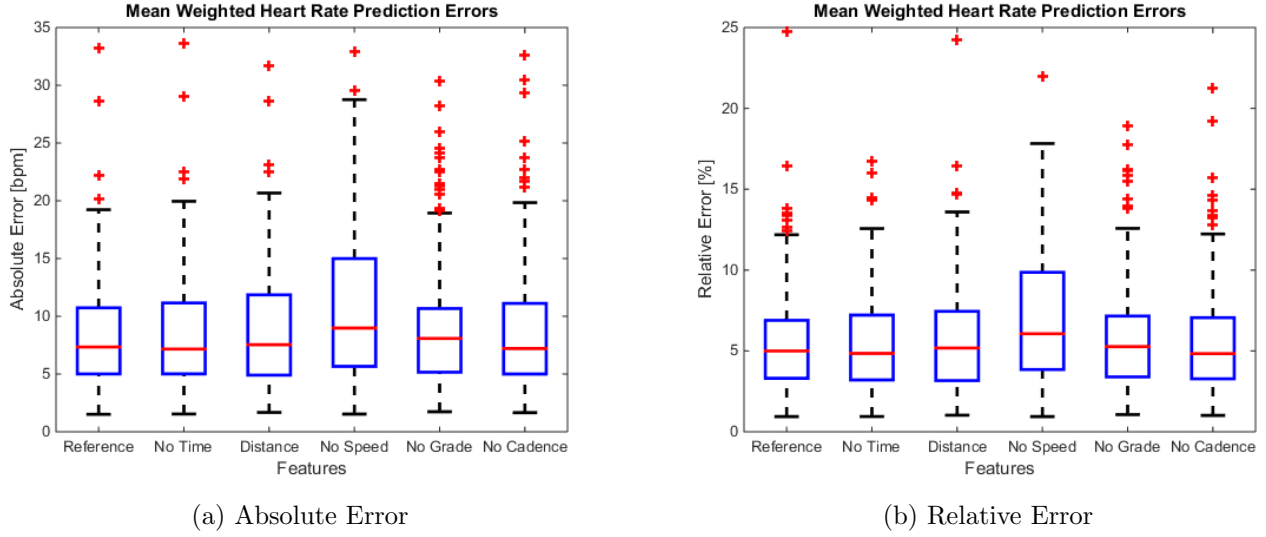


Figure 6.11

For our CRF model, we generally used time, speed, grade, and cadence values during a workout session as input observations. To study the impact of each of those features, we compared the prediction errors when varying the selected features. The following six alternatives were tested:

Reference: {Time, Speed, Grade, Cadence}

No Time: {Speed, Grade, Cadence}

Distance: {Distance, Speed, Grade, Cadence}

No Speed: {Time, Grade, Cadence}

No Grade: {Time, Speed, Cadence}

No Cadence: {Time, Speed, Grade}

Most tests, show only very slight differences. The main observations are:

1. The speed clearly takes an important role in the model, as the 25th and 75th percentile intervals significantly increase without it.
2. The median and 25th percentile values are slightly better if the time is not used as an observation. We suspect, this mainly affects long workouts in the test set, where no similarly long workout was present in the training set, which could lead to potentially undefined behavior.

3. Using the distance instead of the time, is slightly worse. This is possibly because the distance measure fails to accurately reflect pauses during workouts.
4. We see the number of outliers significantly rise without the grade, which reflect the workouts with a lot of elevation changes. The overall effect is very small, because most workouts in the data set are rather flat.
5. The removal of the cadence as well shows only minor effects on the end prediction. This is a confirmation that the freely chosen cadence during a workout session shows very low variation and is therefore suboptimal as a heart rate indicator. (see Section 2.5)

Model Order

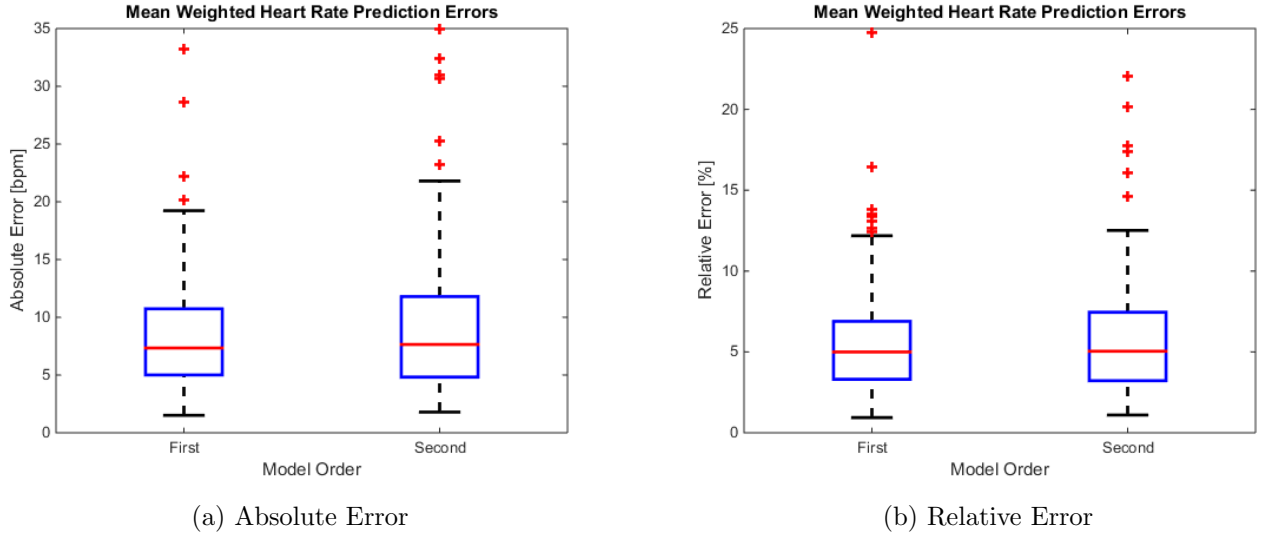


Figure 6.12

For the CRF model, we have not precisely specified a method of modelling the observed heart rate adaption delay (Section 5.1.2). One straight forward way to to this, is to increase the order of the model, so that a prediction at point t not only includes observations from time t but also the last ones at $t - 1$. We call this a second order linear-chain CRF with the following observations:

$$\{\text{time}, \text{speed}_t, \text{grade}_t, \text{cadence}_t, \text{speed}_{t-1}, \text{grade}_{t-1}, \text{cadence}_{t-1}\}$$

For the first data point, the current observation is used twice, as no previous one is available.

As seen in Figure 6.12, the prediction error does however not decrease for the second order model, in comparison to the first order model. One main problem with this approach on the Endomondo data set is, that the temporal distance between multiple observations is not fixed and therefore the previous observation values are suboptimal representatives as they might be spread out over the delay interval. (or even before that)

Temporal Prediction

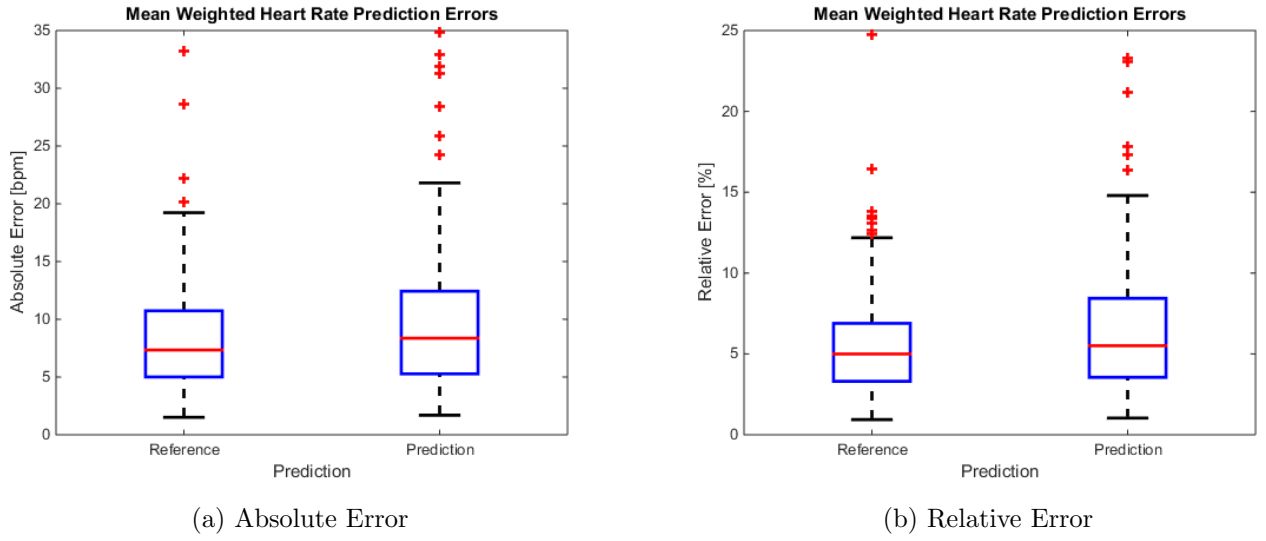


Figure 6.13

Similarly to the second order model, we also want to test, if the proposed CRF model can be used to temporally predict the next heart rate value at point $t + 1$, given only the observations at point t .

Figure 6.13, shows the comparison of the mean prediction error for the reference model and the temporal prediction model, that only uses the observations of the previous data point to infer the next heart rate.

Although the number of outliers and the median prediction errors increase, the CRF model is still suitable for temporal prediction in most cases, as the 75th percentile is only slightly above 10 bpm.

Individual Dataset

Apart from the presented experiments on the Endomondo test set data, we also compare the model prediction error on our individual collection. It is crucial to keep in mind, that the individual collection data is generally of much better quality and has smaller and constant time intervals between two data points (i.e. 5 s). We would therefore expect generally lower errors and a better prediction quality.

Due to a lack of complete workouts in the individual collection, we train the CRF model on 2 workouts of a subject and test it based on a third one. Note that the size of the training size is significantly below our usually used one and the result is consequently bad as seen in Figure 6.14, which shows the complete distribution of errors for the weighted heart rate prediction.

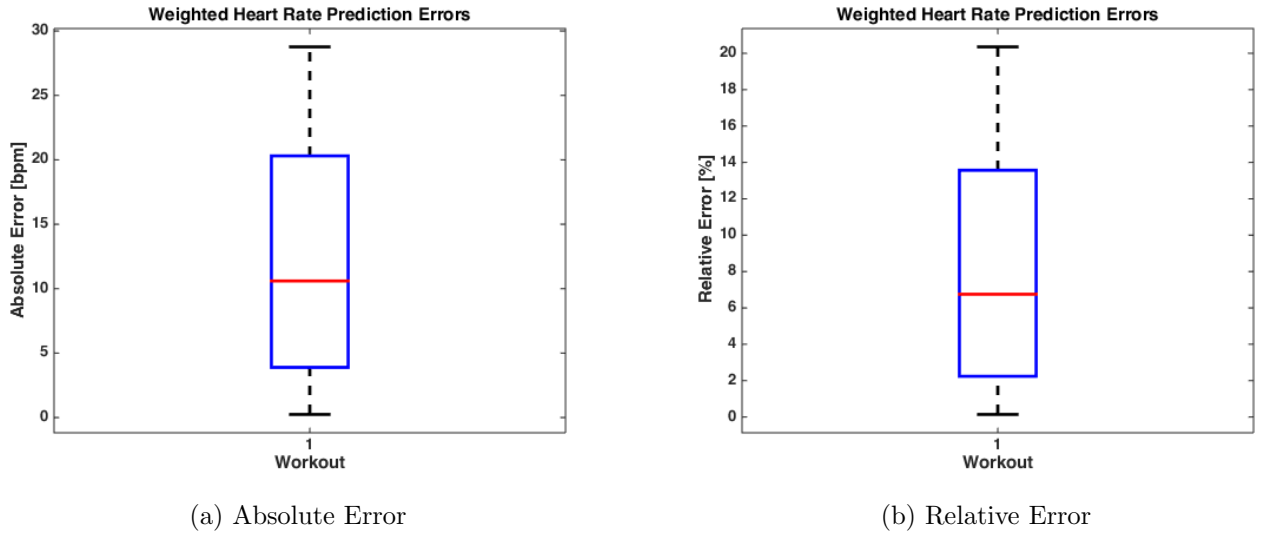


Figure 6.14: Complete boxplot of errors in single test workout of the heart rate model after training on two workouts of a subject in the individual collection.

We learn, that the number of workouts in the individual collection is too low to train our heart rate model sufficiently. The results from the Endomondo collection however, encourage the usability and prediction accuracy of the proposed model.

6.2 Fitness

In Section 5.3, we motivated the use of certain properties in the learned CRF model weights as indicators of an increasing fitness. These properties are partly motivated by literature and weight interpretation; and partly by the assumption, that the fitness level increases steadily over the course of many workouts.

However, for people that already have a very high fitness level, we do not assume, that the level will actually increase significantly with further training. Additionally, we cannot make any statements about decreasing fitness. Most importantly, a non-increasing fitness level based on our properties does not imply a declining fitness level.

With those remarks in mind, we define the previously mentioned improvement function $\text{improvement}(\lambda^{old}, \lambda^{new}) : \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{B}$ as follows:

function $I = \text{IMPROVEMENT}(\lambda^{old}, \lambda^{new})$

properties $\leftarrow \{$	\triangleright Store all property booleans
$\max \lambda_e^{new} < \max \lambda_e^{old},$	
$\text{std}(\lambda_e^{new}) < \text{std}(\lambda_e^{old}),$	
$\text{mean}(b_e^{new}) < \text{mean}(b_e^{old}),$	
$\text{mean}(\lambda_{s,3}^{new}) < \text{mean}(\lambda_{s,3}^{old}),$	
$\ \lambda^{new}\ < \ \lambda^{old}\ ,$	
$\ \lambda_s^{new}\ < \ \lambda_s^{old}\ ,$	
$\}$	
$I \leftarrow \{i \mid \text{properties}_i = \text{true}\} \geq b$	\triangleright At least b of them must be true

The algorithm calculates six weight properties that were discussed to have an impact on the fitness level of a subject. If at least b of these properties increased between the old and new model weights, we assume, the fitness level has risen. The default configuration is $b = 6$, as motivated by the evaluation in the next section.

The choice of the number of properties that have to rise (b), ensures that a certain confidence can be established, that the fitness level actually has risen.

6.2.1 Evaluation

The evaluation of the proposed relative fitness measure is hard. On one hand, we do not have any real ground truth at hand, that would help us validate if our prediction is correct. On the other hand, the measure is highly based on the basic assumption that the fitness level improves with continuous workouts.

Therefore we propose the following method to validate that the measure is reasonable:

1. Select two non-overlapping random subsets of 10 workouts from the same

Case	Occurrences [%]		
	$b = 4$	$b = 5$	$b = 6$
$I = \text{false} \ \& \ \lambda$ not in order	20	23	40
$I = \text{false} \ \& \ \lambda$ in order	34	34	39
$I = \text{true} \ \& \ \lambda$ not in order	29	25	6
$I = \text{true} \ \& \ \lambda$ in order	17	18	15

Table 6.2: Number of cases for possible outcomes in the evaluation of the fitness measure. With $b = 6$, the results seem most plausible as less fitness increases are detected when workout sets are not in the correct temporal order than when they are.

subject

2. Train the CRF model on both subsets and retrieve the feature weights
3. Assign the weights to λ^{old} and λ^{new}
4. Calculate $I = \text{improvement}(\lambda^{old}, \lambda^{new})$
5. Register the value of I and the original temporal order of the subsets
6. Discuss the number of cases where I was true when the second subset was from later workouts (and therefore would represent a better fitness based on our assumption) or not.

We performed 1000 such random comparisons and list the results in Table 6.2. As we cannot really make a statement about the fitness level decreasing, we still consider this to happen significantly less often for the workout sets we gathered from Endomondo. When selecting a final model parameter for the bound b of feature weight properties that have to decrease for the fitness level to be considered increasing, we expect:

1. The “false-positive” rate to be relatively low. These are the cases where $I = \text{true}$ but the order of used workout sets suggests otherwise, as they came in reverse temporal order.
2. The “true-positive” rate to be higher but still not exceptionally high, as the fitness level should not increase too often.

We use the terms “false-positive” and “true-positive” in quotation marks, as our data does not contain enough information to validate if a claim is actually true or false, but only general clues that let us assume one or the other.

Based on the evaluation results, a selection of $b = 6$ seems the best fit, implying that every single one of the six proposed feature weight properties have to decline in order for a workout set to be rated as coming from a fitter subject.

Discussion

In this chapter, we summarize the findings and results of this thesis and give an outlook on possible extensions and future work.

The main contributions of this thesis are:

1. Two workout data sets: a small one consisting of individually collected data using off-the-shelf smart phone, and a large one with publicly available workout data from the internet.
2. Two heart rate models to predict the heart rate during exercise using observations of multiple main influences on the energy consumption.
3. A fitness measure that reflects when the fitness level of a subject increases.

In the following sections, we discuss possible further uses for our data sets, as well as the main findings and possible applications of the proposed models.

7.1 Data Sets

Our large set of online workouts from Endomondo, bears a lot of potential in various applications. Each workout, consists of multiple observation features continuously registered over the entire time period of a workout session and additional metadata. The workouts were gathered per user and therefore, the data sets contains multiple workouts from the same users. With some filtering and classification procedures in place, the entire data set can efficiently be reduced to a subset of ones liking. We can think of other applications for the same data in e.g., for identifying workout patterns, common workout tracks or investigating workout schedules.

Overall, the quality of the workout data is quite irregular and it is a challenging task to correctly distinguish workouts with high quality data from instances with some flaws that might skew some calculations. In our heart rate model evaluation, this shows in many prediction error outliers for workouts of mediocre data

quality. A possible future extension of the current model would be an extensive evaluation based on data that was gathered under precisely controlled conditions to allow for statements about its best possible fit. On the other hand however, the data from Endomondo represents the average data quality one can expect when data is gathered, using off-the-shelf smart phones, by users all around the world. As discussed in the next section, the performance of the model on this data set is still high and its predictions well usable in practice.

7.2 Heart Rate Model

We proposed two ways to model the heart rate based on key features of energy consumption during workouts.

The first approach using a linear least squares model shows some trade-off between being simplistic and allowing for good predictions. Both models were constructed with the constraint that the heart rate is not used as an input. It could however be interesting to evaluate the performance of a linear model with the sole aim of temporal prediction of the heart rate. Such a model could use the current heart rate and observation feature values to predict the heart rate in the near future. We expect a linear model to be better at this constrained task than the global prediction we aimed for.

The more sophisticated second heart rate model based on a linear-chain Conditional Random Field (CRF), shows average prediction errors of around 7 bpm, corresponding to a relative error of 5%. 75% of all tested workouts showed average errors of less than 11 bpm ($\approx 7\%$). Various researchers investigated different workout intensities and their impact on the body. Based on those studies, multiple heart rate zone models have been proposed to suggest a range the heart rate should stay in during an exercise to achieve the intended training effect. [58] Those zones are typically formulated as percentages of HR_{\max} . For example in [59], McArdle et al. propose exercising at a heart rate in the range from 70% to 80% of HR_{\max} to improve aerobic fitness. Depending on the actual value of HR_{\max} , this range comprises about 17 to 22 bpm which is significantly higher than the error we expect from our heart rate prediction. We conclude, that the error of a heart rate prediction by our CRF model will in almost all cases not lead to a change in the heart rate zone and therefore training intensity.

We encourage the use of our proposed linear-chain CRF model to predict future changes in the heart rate based on changing workout features like speed or grade and as a tool to better understand the impact of different features on the resulting heart rate and exercise intensity. We also showed, that valuable information about a subject's fitness level can be extracted from a trained workout model. An extensive evaluation of the proposed fitness improvement function, using an alternative fitness measure like $VO_2 \max$ as reference, would be an

interesting next step.

7.3 Future Work

The BeatRunner application could be extended to make use of our proposed CRF heart rate model to predict the impact of feature changes on the heart rate and find a cadence that would be more suitable to keep the heart rate close to the defined goal. We remark, that we suspect that the entire approach of using the music beat to control the heart rate during exercise does not work as good as might be expected. (see Section 2.5) It would however be interesting to see a separate evaluation of this underlying assumption.

We compared the prediction error of our CRF model for the disciplines running, mountain biking and cycling. The results suggest, that the model best fits the originally intended discipline of running. It would be interesting to study more aspects of other sport disciplines to either improve the current model to fit all of them or build separate model for each sport.

A linear-chain CRF is the simplest version of a Conditional Random Field model. As a possible extension to this thesis, we could investigate different kinds of CRF models and especially other feature functions, that may reflect even more fine grained aspects of a training session.

Bibliography

- [1] ETH Zurich: Declaration of Originality, Computer Science Department, ETH Zurich. http://www.inf.ethz.ch/content/dam/ethz/special-interest/infk/departement/Images%20and%20Content/Studies/Forms%20and%20Documents/2014-04-16_Declaration-Originality_en.pdf Accessed: 2015-04-27.
- [2] Huonder, V.: Beatrunner - an android application to enhance a workout through music (6 2014)
- [3] Graf, M.: Using music to enhance your running (1 2014)
- [4] Keilin, D.: On cytochrome, a respiratory pigment, common to animals, yeast, and higher plants. *Proceedings of the Royal Society of London B: Biological Sciences* **98**(690) (1925) 312–339
- [5] K., L.: Über die pyrophosphatfraktion im muskel. *Naturwissenschaften* **17**(31) (1929) 624–625
- [6] Rich, P.: The molecular machinery of keilin’s respiratory chain. *Biochemical Society Transactions* **31**(6) (2003) 1095–1106
- [7] Margaria, R.: *Atti accad. Naz. Lincei Memorie* **7** (1938) 299–368
- [8] Margaria, R., Cerretelli, P., Aghemo, P., Sassi, G.: Energy cost of running. *Journal of Applied Physiology* **18**(2) (1963) 367–370
- [9] Davies, C.T., Sargeant, A.J., Smith, B.: The physiological responses to running downhill. *European Journal of Applied Physiology and Occupational Physiology* **32**(3) (1974) 187–194
- [10] Minetti, A.E., Moia, C., Roi, G.S., Susta, D., Ferretti, G.: Energy cost of walking and running at extreme uphill and downhill slopes. *Journal of Applied Physiology* **93**(3) (2002) 1039–1046
- [11] Minetti, A., Ardigo, L., Saibene, F.: Mechanical determinants of the minimum energy cost of gradient running in humans. *The Journal of experimental biology* **195**(1) (1994) 211–225
- [12] Davies, C.T.: Effects of wind assistance and resistance on the forward motion of a runner. *Journal of Applied Physiology* **48**(4) (1980) 702–709

- [13] Pugh, L.G.C.E.: Oxygen intake in track and treadmill running with observations on the effect of air resistance. *The Journal of physiology* **207**(3) (1970) 823–835
- [14] Pugh, L.G.C.E.: The influence of wind resistance in running and walking and the mechanical efficiency of work against horizontal or vertical forces. *The Journal of Physiology* **213**(2) (1971) 255–276
- [15] Cavagna, G., Franzetti, P., Heglund, N., Willems, P.: The determinants of the step frequency in running, trotting and hopping in man and other vertebrates. *The Journal of Physiology* **399**(1) (1988) 81–92
- [16] Cavagna, G., Willems, P., Franzetti, P., Detrembleur, C.: The two power limits conditioning step frequency in human running. *The Journal of physiology* **437**(1) (1991) 95–108
- [17] Staum, M.J.: Music and rhythmic stimuli in the rehabilitation of gait disorders. *Journal of music therapy* **20**(2) (1983) 69–87
- [18] Oliver, N., Flores-Mangas, F.: Mptrain: A mobile, music and physiology-based personal trainer. In: *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services. MobileHCI '06*, New York, NY, USA, ACM (2006) 21–28
- [19] de Oliveira, R., Oliver, N.: Triplebeat: Enhancing exercise performance with persuasion. In: *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services. MobileHCI '08*, New York, NY, USA, ACM (2008) 255–264
- [20] Edwards, S.: *The heart rate monitor guidebook to heart zone training. Heart Zones* (2010)
- [21] Gina, K.: 'maximum' heart rate theory is challenged. *New York Times* (2001)
- [22] Fortmann, J., Pielot, M., Mittelsdorf, M., Büscher, M., Trienen, S., Boll, S.: Paceguard: Improving running cadence by real-time auditory feedback. In: *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services Companion. MobileHCI '12*, New York, NY, USA, ACM (2012) 5–10
- [23] Koenig, A.C., Somaini, L., Pulfer, M., Holenstein, T., Omlin, X., Wieser, M., Riener, R.: Model-based heart rate prediction during lokomat walking. In: *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE, IEEE* (2009) 1758–1761
- [24] Xiao, F., Chen, Y., Yuchi, M., Ding, M., Jo, J.: Heart rate prediction model based on physical activities using evolutionary neural network. In:

- Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on, IEEE (2010) 198–201
- [25] Xiao, F., Yuchi, M., Ding, M., Jo, J.: A research of heart rate prediction model based on evolutionary neural network. In: Intelligent Computation and Bio-Medical Instrumentation (ICBBI), 2011 International Conference on, IEEE (2011) 304–307
 - [26] Shvartz, E., Reibold, R.: Aerobic fitness norms for males and females aged 6 to 75 years: a review. *Aviation, space, and environmental medicine* **61**(1) (1990) 3–11
 - [27] Hall, J.E.: *Guyton and Hall textbook of medical physiology*. Elsevier Health Sciences (2010)
 - [28] Fick, A.: Ueber die messung des blutquantums in den herzventrikeln. *Sitz. Physik. Med. Ges* **2** (1870) 16
 - [29] Cooper, K.H.: A means of assessing maximal oxygen intake: correlation between field and treadmill testing. *Jama* **203**(3) (1968) 201–204
 - [30] Léger, L.A., Lambert, J.: A maximal multistage 20-m shuttle run test to predict\ dot vo2 max. *European journal of applied physiology and occupational physiology* **49**(1) (1982) 1–12
 - [31] Uth, N., Sørensen, H., Overgaard, K., Pedersen, P.K.: Estimation of vo2max from the ratio between hrmax and hrrest—the heart rate ratio method. *European journal of applied physiology* **91**(1) (2004) 111–115
 - [32] Fox 3rd, S., Naughton, J.P., Haskell, W.L.: Physical activity and the prevention of coronary heart disease. *Annals of Clinical Research* **3**(6) (1971) 404–432
 - [33] Robergs, R.A., Landwehr, R.: The surprising history of the hrmax= 220-age equation. *J Exerc Physiol* **5**(2) (2002) 1–10
 - [34] Polar Electro: How to Utilize Running Index in your training. http://www.polar.com/us-en/support/How_to_Utilize_Running_Index_in_your_training Accessed: 2015-05-07.
 - [35] Polar Electro: Polar RS400 Online Manual - Polar Running Index. http://www.polar.com/e_manuals/RS400/Polar_RS400_user_manual_English/ch10.html#Polar_running_index Accessed: 2015-05-07.
 - [36] College, O.: *Anatomy & Physiology*. 7.26 edn. OpenStax College (2015)

- [37] Kounalakis, S.N., Nassis, G.P., Koskolou, M.D., Geladas, N.D.: The role of active muscle mass on exercise-induced cardiovascular drift. *Journal of Sports Science and Medicine* **7** (2008) 395–401
- [38] Montain, S.J., Coyle, E.F.: Influence of graded dehydration on hyperthermia and cardiovascular drift during exercise. *Journal of Applied Physiology* **73**(4) (1992) 1340–1350
- [39] Google Inc.: Android Phone Axis. <https://developer.android.com/reference/android/hardware/SensorEvent.html> Accessed: 2015-04-24.
- [40] Google Inc.: Documentation about accuracy value in Android GPS Position. [https://developer.android.com/reference/android/location/Location.html#getAccuracy\(\)](https://developer.android.com/reference/android/location/Location.html#getAccuracy()) Accessed: 2015-04-15.
- [41] Grewal, M., Weill, L., Andrews, A.: *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley (2007)
- [42] Townshend, A.D., Worringham, C.J., Stewart, I.B.: Assessment of speed and position during human locomotion using nondifferential gps. *Medicine and science in sports and exercise* **40**(1) (2008) 124–132
- [43] Imagery, U.S.N., Agency, M.: *Department of Defense World Geodetic System 1984: Its Definition and Relationships With Local Geodetic Systems*. third edition edn. National Imagery and Mapping Agency (2000)
- [44] Berberan-Santos, M.N., Bodunov, E.N., Pogliani, L.: On the barometric formula. *American Journal of Physics* **65** (May 1997) 404–412
- [45] Zhao, N.: Full-featured pedometer design realized with 3-axis digital accelerometer. *Analog Dialogue* **44** (2010) 1–5
- [46] Bluetooth SIG: Bluetooth GATT Running Speed and Cadence Service Specification. https://developer.bluetooth.org/gatt/services/Pages/ServiceViewer.aspx?u=org.bluetooth.service.running_speed_and_cadence.xml Accessed: 2015-04-16.
- [47] Bluetooth SIG: Bluetooth GATT Running Speed and Cadence Characteristics Specification. https://developer.bluetooth.org/gatt/characteristics/Pages/CharacteristicViewer.aspx?u=org.bluetooth.characteristic.rsc_measurement.xml Accessed: 2015-04-16.
- [48] Google Inc.: Google Elevation API Documentation. <https://developers.google.com/maps/documentation/elevation> Accessed: 2015-04-28.
- [49] Efroelicher, V.: Exercise and the heart. *Chest* **91** (1987) 585

- [50] Minetti, A.E., Gaudino, P., Seminati, E., Cazzola, D.: The cost of transport of human running is not affected, as in walking, by wide acceleration/deceleration cycles. *Journal of Applied Physiology* **114**(4) (2013) 498–503
- [51] Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 282–289
- [52] Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Statist.* **37**(6) (12 1966) 1554–1563
- [53] Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In Getoor, L., Taskar, B., eds.: *Introduction to Statistical Relational Learning*. MIT Press (2007)
- [54] Elkan, C.: Log-linear models and conditional random fields (February 2014)
- [55] Sutton, C., McCallum, A.: An introduction to conditional random fields. *Foundations and Trends in Machine Learning* **4**(4) (2012) 267373
- [56] Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* **16**(5) (1995) 1190–1208
- [57] Bertsekas, D.: *Nonlinear Programming*. 2nd edition edn. Athena Scientific (1999)
- [58] Alderson, J., Beashel, P., Taylor, J.: *Advanced studies in physical education and sport*. Nelson Thornes (1996)
- [59] McArdle, W.D., Katch, F.I., Katch, V.L.: *Essentials of exercise physiology*. Lippincott Williams & Wilkins (2006)
- [60] Proprietary, B.S.: *Specification of the Bluetooth[®] System*. Version 4.2 edn. Bluetooth SIG Proprietary (2014)

Theory Details

A.1 Boxplots

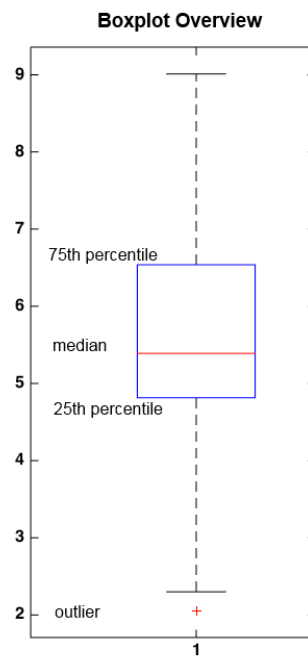


Figure A.1: Example Boxplot showing median, 25th and 75th percentile. The whiskers are bounds to all points considered within the set while outliers are separately drawn as red markers.

More Experiment Results

B.1 Hardware

Two different desktop tower machines running Windows 7 Service Pack 2 (64 bit) were used. Both ran the 64bit version of MathWorks Matlab R2014b. The detailed specifications are listed below.

Machine 1:

Processor Intel i7-3770 @ 3.4 GHz

RAM 16GB

GPU NVIDIA GeForce GTX 660 Ti

Main Disk Samsung 840 Pro Basic 256GB SSD

Machine 2:

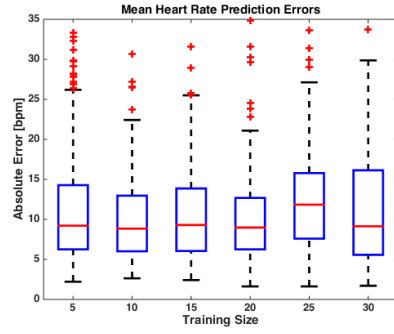
Processor Intel i7-4790K @ 4Ghz

RAM 16GB

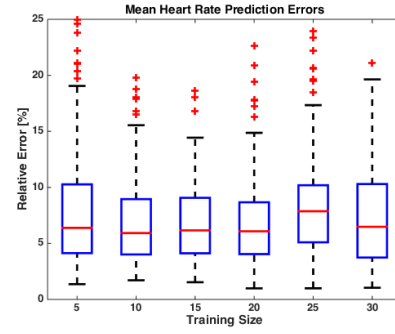
GPU 2x NVIDIA GeForce GTX 780 Ti

Main Disk Samsung 840 Pro 512GB SSD

B.2 Conditional Random Field (CRF)

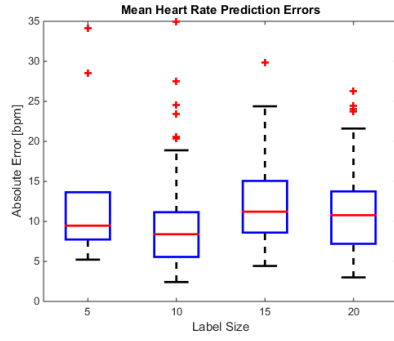


(a) Absolute Error

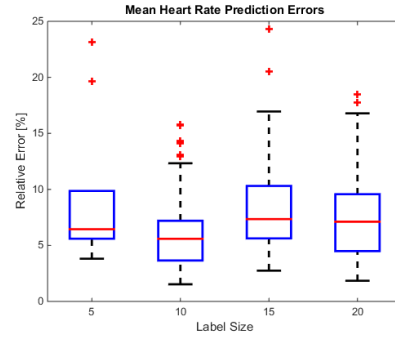


(b) Relative Error

Figure B.1: Varying Training Set Size; raw errors.



(a) Absolute Error



(b) Relative Error

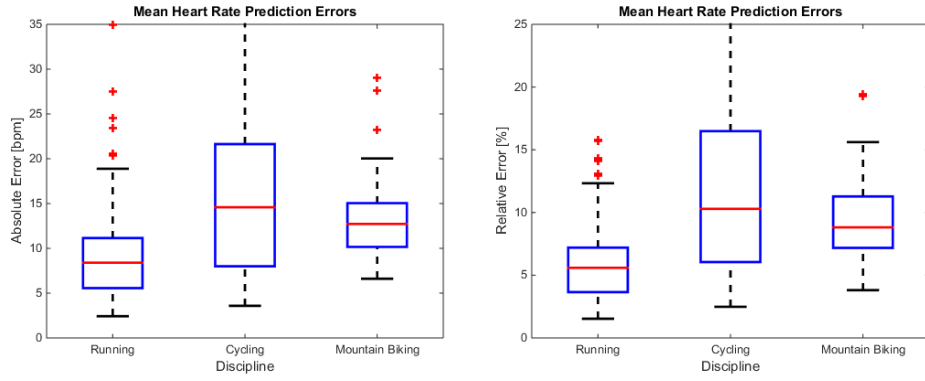
Figure B.2: Varying State Size δ ; raw errors.

(a) Absolute Error



(b) Relative Error

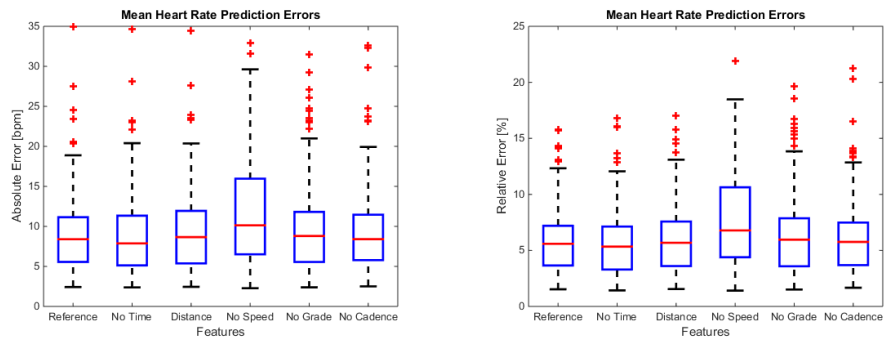
Figure B.3: Comparing Model Order; raw errors.



(a) Absolute Error

(b) Relative Error

Figure B.4: Comparing Disciplines; raw errors.



(a) Absolute Error

(b) Relative Error

Figure B.5: Comparing feature selection; raw errors.

Code Notes

comments about different scripts and implementations provided - endomondo
Scraper - Step Frequency Algorithm

C.1 Google Elevation API

A simple query to the Google Elevation API (see Section 4.2.2) contains a pipe (i.e. |) separated list of `latitude,longitude` tuples and is made via a HTTP GET request to the following URL:

```
https://maps.googleapis.com/maps/api/elevation/json  
?locations=<location string>  
&key=<API Key>
```

The response is a JSON document containing both an elevation in meters above the WGS 1984 reference ellipsoid and a resolution value indicating the maximum distance between data points from which the elevation was interpolated, in meters.

We used the Google Elevation API to find elevation data for all 2199 workouts in the filtered Endomondo data set. Since the API has a usage limit of only 2500 queries and 25 000 positions per day, queries had to be split into multiple days as each workout contains an average of about 290 data points.

C.2 Step Detection Algorithm

Algorithm C.1 Step Detection Algorithm

function $S, C = \text{DETECTSTEPS}(A)$
 S : Timestamps and length tuples $(s_j, \Delta s_j)$ of detected steps
 C : Vector of time and corresponding cadences
 A : Acceleration tuples $(t_i, a_i) \forall i \in \{1 \dots n\}$
 $\tau \leftarrow \tau^{\text{minw}} \cdot \min_i(a_i) + (1 - \tau^{\text{minw}}) \cdot \max_i(a_i), \tau_{\text{old}} \leftarrow \tau$ \triangleright Threshold
 $a^{\text{min}} \leftarrow \infty$
 $a^{\text{max}} \leftarrow -\infty$
 $i \leftarrow 1$ \triangleright Sample counter initialization
 $j \leftarrow 1, s'_{\text{old}} \leftarrow t_0$ \triangleright Step counter and time initialization
 $k \leftarrow 0$ \triangleright Cadence index
 $sc_{\text{val}} \leftarrow 0, sc_{\text{inv}} \leftarrow 0$ \triangleright Successive valid and invalid step counts
for all $(t_i, a_i) \in A$ **do**
 $a^{\text{min}} \leftarrow \min(a_i, a^{\text{min}})$
 $a^{\text{max}} \leftarrow \max(a_i, a^{\text{max}})$
if $a_i \leq \tau \wedge a_{i-1} > \tau_{\text{old}}$ **then** \triangleright Threshold passed from top to bottom
 $s'_{\text{new}} \leftarrow t_i$ \triangleright Assign temporary step information
 $\Delta s' \leftarrow s'_{\text{new}} - s'_{\text{old}}$
if $\Delta s^{\text{min}} \leq \Delta s'_j \leq \Delta s^{\text{max}}$ **then** \triangleright Step length is valid
 $(s_j, \Delta s_j) \leftarrow (s'_{\text{new}}, \Delta s')$
 $j \leftarrow j + 1$
 $sc_{\text{val}} \leftarrow sc_{\text{val}} + 1, sc_{\text{inv}} \leftarrow 0$
 $s'_{\text{old}} \leftarrow s'_{\text{new}}$
if $sc_{\text{val}} \geq c^{\text{update}}$ **then**
 $c \leftarrow 60 \cdot w / (s_j - s_{j-w+1})$ \triangleright Update cadence
 $C_k \leftarrow (s_j, c)$
 $k \leftarrow k + 1$
else \triangleright Step invalid
 $sc_{\text{val}} \leftarrow 0, sc_{\text{inv}} \leftarrow sc_{\text{inv}} + 1$
if $\Delta s' > \Delta s^{\text{max}}$ **then** \triangleright Step length too high
 $s'_{\text{old}} \leftarrow s'_{\text{new}}$
if $sc_{\text{inv}} \geq c^{\text{reset}}$ **then** \triangleright Reset Cadence to 0
 $C_k \leftarrow (t_i, 0)$
 $k \leftarrow k + 1$
if $i = \tau^w$ **then** \triangleright Update Threshold
 $\tau \leftarrow \tau^{\text{minw}} \cdot a^{\text{min}} + (1 - \tau^{\text{minw}}) \cdot a^{\text{max}}$
 $i \leftarrow 1$
 $i \leftarrow i + 1$

Data Set Details

D.1 Endomondo

D.1.1 Profile Information

Additionally to the downloaded workout data, we gathered general profile information for each user. Listing D.1 shows all gathered profile data for user 1642, as an example.

```
{
  "personal_best": {
    "Cycling_sport": {
      "10_miles": "25m:41s",
      "100_km": "3h:00m:16s",
      "20_km": "32m:35s",
      "50_km": "1h:24m:45s",
      "50_miles": "2h:24m:05s",
      "One_hour": "36.08km"
    },
    "Running": {
      "1_km": "3m:24s",
      "1_mile": "6m:22s",
      "10_km": "45m:28s",
      "12-minute_test": "2.79km",
      "3_km": "13m:00s",
      "3_miles": "21m:31s",
      "5_km": "22m:18s",
      "Half_marathon": "1h:44m:55s",
      "Marathon": "4h:04m:08s",
      "One_hour": "12.29km"
    }
  },
  "profile_id": "1642",
  "stats": {
    "calories": {
      "Burgers_burned": "1881",
      "Calories_burned": "1015740kcal"
    },
    "distance": {
      "Total_distance": "17006.61km",
```

```

        "Trips_around_the_world": "0.424",
        "Trips_to_the_Moon": "0.044"
    },
    "duration": {
        "Total_duration": "2m:1d:10h"
    },
    "pace": {
        "Average_pace": "5m:07s"
    },
    "speed": {
        "Average_speed": "11.72km/h"
    },
    "workouts": {
        "Total_workouts": "1667"
    }
},
"subject": {
    "Birthday": "Jul_30,_1956",
    "Country": "Denmark",
    "Favorite_Sport": "Running",
    "Postal_Code": "8270",
    "Sex": "Male"
}
}

```

Listing D.1: Example profile data of Endomondo user 1642.

D.1.2 JSON Workout Data

The JSON data in Listing D.2 is an extract of an Endomondo mountain bike exercise session. Both the **distance** and **duration** field are accumulative and can be used as the x-axis of a plot.

```

{
  "data": [
    {
      "values": {
        "hr": 141.40741,
        "distance": 3.3843606,
        "duration": 502050,
        "cad": 21.074074,
        "speed": 22.342001,
        "alt": 149.21478
      },
      "lng": -4.609755640849471,
      "lat": 37.431532703340054
    }, ...
  ], ...
}

```

Listing D.2: Example JSON data of an Endomondo workout

D.1.3 Detailed Statistics

```

{
  "age": {
    "avg": 40.25,
    "max": 77.00,
    "min": 0,
    "n": 2781,
    "n/a": 659,
    "std": 10.02,
    "var": 100.44
  },
  "bmi": {
    "avg": 24.35,
    "max": 37.36,
    "min": 0,
    "n": 339,
    "n/a": 3101,
    "std": 2.98,
    "var": 8.91
  },
  "collection": {
    "avg": 62.55,
    "max": 1486,
    "min": 0,
    "n": 3360,
    "std": 143.69,
    "var": 20646.24
  },
  "collection_%": {
    "avg": 16.50,
    "max": 119.03,
    "min": 0,
    "n": 3360,
    "std": 27.45,
    "var": 753.63
  },
  "gender": {
    "Female": 443,
    "Male": 2805,
    "n/a": 112
  },
  "height": {
    "avg": 1.77,
    "max": 2.03,
    "min": 0,
    "n": 2229,
    "n/a": 1211,
    "std": 0.09,
    "var": 0.01
  },
  "speed": {
    "avg": 11.15,
    "max": 2249.66,
    "min": 0,
    "n": 3342,
    "n/a": 18,
    "std": 39.62,
    "var": 1569.35
  },
  "weight": {
    "avg": 80.08,
    "max": 226,
    "min": 0,
    "n": 352,
    "n/a": 3088,
    "std": 19.08,
    "var": 364.05
  },
  "workouts": {
    "avg": 443.17,
    "max": 3900,
    "min": 0,
    "n": 3360,
    "std": 408.63,
    "var": 166974.62
  }
}

```

Listing D.3: Complete Data Set Statistics

```

{
  "age": {
    "avg": 42.33,
    "max": 64.00,
    "min": 0,
    "n": 126,
    "n/a": 35,
    "std": 8.97,
    "var": 80.43
  },
  "bmi": {
    "avg": 24.26,
    "max": 28.91,
    "min": 0,
    "n": 35,
    "n/a": 126,
    "std": 2.54,
    "var": 6.45
  },
  "collection": {
    "avg": 12.29,
    "max": 177,
    "min": 0,
    "n": 156,
    "std": 26.53,
    "var": 703.77
  },
  "collection_%": {
    "avg": 2.14,
    "max": 33.71,
    "min": 0,
    "n": 156,
    "std": 4.86,
    "var": 23.58
  },
  "gender": {
    "Female": 6,
    "Male": 149,
    "n/a": 1
  },
  "height": {
    "avg": 1.80,
    "max": 2.00,
    "min": 0,
    "n": 129,
    "n/a": 32,
    "std": 0.07,
    "var": 0.01
  },
  "speed": {
    "avg": 13.39,
    "max": 27.66,
    "min": 0,
    "n": 156,
    "std": 6.39,
    "var": 40.85
  },
  "weight": {
    "avg": 79.14,
    "max": 103,
    "min": 0,
    "n": 35,
    "n/a": 126,
    "std": 10.81,
    "var": 116.81
  },
  "workouts": {
    "avg": 815.66,
    "max": 4191,
    "min": 0,
    "n": 156,
    "std": 615.91,
    "var": 379344.65
  }
}

```

Listing D.4: Test Set Statistics

Glossary

E.1 Acronyms

ATP adenosine triphosphate 3	GATT Generic Attribute Profile 25
BFGS BroydenFletcherGoldfarbShanno 44	GPS Global Positioning System 9, 17, 19, 20
BMI Body mass index 34	HMM Hidden Markov Model 42
bpm beats per minute 5, 6, 8, 9, 17, 69	HTML Hypertext Markup Language 30
CRF Conditional Random Field iv, 42, 46, 56, 69, B-1	HTTP Hypertext Transfer Protocol 30
CSV Comma-Separated Values 18	URL Uniform Resource Locator 28
DOM Document Object Model 30	WGS 1984 World Geodetic System 1984 [43] 20

E.2 Glossary

VO₂ max The maximal rate of oxygen consumption during incremental exercise. This value has been shown to reflect the aerobic physical fitness of a subject 7, 8, 46, 69

Android Android is a mobile operating system developed by Google 16, 26

BeatRunner An Android app for fitness tracking that helps to improve the workout using music. See Section 2.5 1, 3, 6, 8–10, 18, 27, 70

Bluetooth A wireless standard built for short ranges and available in most modern mobile devices. Bluetooth is specified by Bluetooth SIG Proprietary in [60] (previously under IEEE 802.2.15) 16–18

Goolge Earth A freely available map software developed by Google Inc. offering satellite imagery of the whole world 20

lokomat A medical treadmill-like device used for rehabilitation of patients after a stroke, spinal cord injury or traumatic brain injury. Typically used at very low speeds while precisely monitoring the patient 6

python Python is an interpreted high-level programming language 30

spirometer A device for measuring the volume of air inhaled and exhaled by the lungs
7

Wi-Fi A widely used wireless computer networking technology 10