RANGE PREDICTION FOR E-BIKES

BACHELOR THESIS

Lukas Gebhard
Matrikel: 67337
born September 21, 1992

Supervisor:
Prof. Dr. Hermann de Meer
Chair of
Computer Networks and Computer Communications
Universität Passau

Begin: December 21, 2015
Submission: March 7, 2016
Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. This paper was not previously presented to another examination board.

Passau, March 7, 2016

Lukas Gebhard
Abstract

Thanks to their affordability and practicality, electric bicycles (e-bikes) are becoming popular, especially in urban areas. They are a zero-emission alternative to cars and therefore have a high potential to mitigate climate change. However, range anxiety can be a significant barrier to the adoption of electric vehicles. To address this challenge, in this paper we focus on how an e-bike’s remaining range can be accurately predicted. Using real data from an e-bike field trial, combined with OpenStreetMap data, we evaluate several range prediction methods that take riding behaviour and route characteristics into account. Surprisingly, we find that predicting range based on energy consumption for historical trips by that e-bike works just as well as more complex methods that include additional information, such as route being travelled. Our findings also reveal which additional hardware and sensors e-bike manufacturers should provide in the future to make it easier to implement on-board range prediction. To the best of our knowledge, this is the first study of range prediction specifically for e-bikes.
1 Introduction

According to the fifth assessment report of the Intergovernmental Panel on Climate Change (IPCC), the global transportation sector accounts for 14 percent of all anthropogenic greenhouse gases \(^{(10)}\). Electric vehicles can play a key role in mitigating this problem, especially as electricity grids worldwide transition to renewable energy sources. In particular, electric bicycles (e-bikes) are a rapidly-growing mode of urban transportation, with over 200 million on the road in China \(^{(18)}\) and annual sales expected to grow from 32 million in 2014 to 40 million in 2023 \(^{(11)}\). E-bikes provide a clean (zero-emission and zero-carbon) and relatively inexpensive option for urban transportation as compared to cars, while being less physically demanding and allowing greater range and speed than regular bicycles.

The problem we study in this paper is how to predict an e-bike’s remaining range. This is an important problem because range anxiety, i.e., the fear of running out of battery with no place to recharge it before reaching the destination, is known to be a major barrier to the adoption of electric vehicles (see, e.g., \(^{(3)}\)). Furthermore, in some parts of the world, e-bikes have been displacing cars and motorcycles, and therefore are used on long trips \(^{(1)}\). Some e-bikes allow the rider to pedal, but the combination of weight (e-bikes can be much heavier than traditional bikes), usage patterns (some owners use their e-bikes for shopping trips and therefore carry additional weight), and drag resistance from the motor can make pedalling unpleasant or even impractical.

Currently, most e-bike manufacturers publish the maximum range of their models, but it is not clear how this number is derived or how accurate it is. Furthermore, while most e-bikes are equipped with a digital display such as that shown in Figure 1.1, the display usually shows battery voltage, which is only a rough approximation of the remaining range. Thus, a more accurate and principled approach is required and would serve to mitigate some barriers to the adoption of e-bikes.

There has been a great deal of work on range estimation for electric cars (details in chapter 2), but we are not aware of any published research specifically for e-bikes, which raise new challenges. For one, to keep prices low, e-bikes are not equipped with sophisticated sensors and electronics, and any additional hardware required for on-board range prediction must be inexpensive. For another, since e-bike batteries are smaller, to reduce power consumption, it may not be possible to collect sensor data such as speed and acceleration as frequently as is done by automobile control units. Also, some of the model variables are likely to be different: e-bike range is very likely to be affected by the rider’s weight and pedalling intensity, whereas an electric car’s range is influenced by the use of heating, defrosting and air conditioning (among other things). Thus, while e-bikes are mechanically simpler than cars, we may have both less data as well as less
CHAPTER 1. INTRODUCTION

Figure 1.1: The digital display of an eProdigy e-bike, displaying speed, mileage, time and battery voltage.

accurate data to work with. Last but not least, one can ride an e-bike off-roads, without motor assistance or just push it while walking, which proves challenging when it comes to data processing.

Towards the goal of practical and accurate range estimation for e-bikes, we make the following contributions:

1. We analyze data from a fleet of 31 sensor-equipped e-bikes, combining it with OpenStreetMap [13] data to derive features that affect range.

2. We test two data-driven range prediction approaches on the above datasets: a simple model based on riding history and a more complex regression model that considers the characteristics of the anticipated route. Surprisingly, we find that both approaches have similar predictive power.

3. We provide recommendations to e-bike manufacturers about the additional sensing hardware that is required to implement our range prediction approaches.

The remainder of this paper is organized as follows. chapter 2 discusses related work on range prediction for electric vehicles; chapter 3 defines the specific problems we want to solve; chapter 4 discusses the required data processing steps; chapter 5 presents our range prediction techniques and evaluates their performance; and chapter 6 concludes the paper with lessons learned and directions for future work.
2 Related work

While we are not aware of research on range estimation for e-bikes, which are still a relatively new technology, there is a long history of work on range estimation for cars [19]. In particular, recent work on range estimation for electric cars can be classified into model-based, data-driven, and combined techniques.

2.1 Model-based approach

Model-based approaches employ physical modelling of the vehicle and the environment. For example, Hayes et al. have developed a general powertrain model (Figure 2.1) with various parameters for different types of vehicles [8]. Then, given the planned route, they incorporate route topography into the model based on data from Google Earth. Oliva et al. propose another model-based approach employing Markov chains and particle filters to account for non-deterministic factors such as traffic situation and driving style [12]. To increase prediction accuracy, one can continue to model the vehicle and its environment in-depth. For instance, Valentina et al. present a detailed model of an electric vehicle’s air conditioning system [23].

2.2 Data-driven approach

In contrast, data-driven approaches do not require much knowledge of the internals of the vehicle, and aim to identify and model the factors affecting range from observable data. Vehicle sensor data can easily be accessed via the standardized Onboard Diagnostics port while smartphones are equipped with GPS sensors, accelerometers and gyroscopes. To give an example, Bolovinou et al. present a regression-based online learning system that dynamically updates model parameters while driving [2]. As main features they choose both long-term as well as short-term past energy consumption. Doing so, they indirectly account for environment-related factors (e.g., traffic jam, rainfall) as these are partly captured by short-term consumption measurements. What route-related features are concerned, elevation and the expected speed profile are considered. Evaluating their system on a total of 2000 km test drives, they retain a root mean absolute error as low as 1.95 km.

A rather novel method is to gather and analyze crowd-sourced sensor data. This method, also known as participatory sensing, is gaining more and more popularity as it allows to reveal collective behaviour. In case of range prediction, this might be specific driving patterns, or speed profiles. In practice, the collected data are often uploaded to a
specialized cloud platform. For instance, CloudThink [4] helps to manage, analyze and access driving data in real-time.

To go more into detail, we exemplarily want to take a closer look at the comprehensive approach suggested by Tseng et al. [20]. It considers three feature classes: Driver-, vehicle- and route-dependent features. For each triple (driver, vehicle, route), a speed profile can be derived. The collected speed profiles are stored in a cloud database. One can then fit an energy consumption model and use the crowd-sourced speed profiles as follows: Let us consider a specific (driver, vehicle, route) triple \((D, V, R)\). Note that, for simplification, route segmentation is omitted at this point. We distinguish two cases: If there is already a speed profile for \((D, V, R)\) in the cloud, we can directly use it for range estimation. Otherwise, we proceed as follows: For each (driver, vehicle) pair \((D_i, V_i)\), \(i = 1, \ldots, k\), we search for all routes \(R_1, \ldots, R_l\) so that there are speed profiles both for \((D, V, R_j)\) and \((D_i, V_i, R_j)\), \(j = 1 \ldots l\). Now Tseng et al. suggest a correlation metric \(\chi_{(D,V),(D_i,V_i)}\). As an intuition, this metric measures the similarity of \((D, V)\) and \((D_i, V_i)\). That is, \(\chi_{(D,V),(D_i,V_i)}\) tells if \(D\) and \(D_i\) have similar driving habits and if \(V\) and \(V_i\) can be compared in terms of vehicle characteristics. Technically, the metric is derived using Dynamic Time Warping, an algorithm that takes two time series as an input and outputs a distance value. Finally, as we calculated all \(\chi_{(D,V),(D_i,V_i)}, i = 1 \ldots k\), we select

\[
(D', V') := \arg\min_{(D_i, V_i)=1 \ldots k} \chi_{(D,V),(D_i,V_i)}
\]

and use \((D', V', R)\) to predict the range of \((D, V, R)\). To illustrate this, here is an
example provided by Tseng et al. Three (driver, vehicle) pairs \( (D_1, V_1), (D_2, V_2), (D_3, V_3) \) are given. Each of them recorded a speed profile on a test route \( R \), as plotted in Figure 2.2.

Figure 2.2: Speed profiles on route \( R \). Source: Tseng et al. [20]

One observes that the speed profile of \( (D_2, V_2, R) \) is closer to \( (D_1, V_1, R) \) than that of \( (D_3, V_3, R) \). This is reflected by the correlation metrics obtained when applying Dynamic Time Warping:

\[
\chi(D_1, V_1), (D_2, V_2) = 1.1385 < 1.3883 = \chi(D_1, V_1), (D_3, V_3)
\]

Testing their system on a Nissan Leaf, Tseng et al. measure high prediction accuracy throughout the test drive shown in Figure 2.3 whereas the onboard meter differs from the actual distance-to-empty remarkably. However, it is important to note that this approach does not account for aspects like temperature, rainfall, heating and air conditioning. So instead of \( (D, V, R) \) triples it might be more reasonable to look at \( (D, V, R, E) \) combinations where \( E \) represents a class of environment-related features.

### 2.3 Combined approach

Last but not least, one can also fuse power train modelling and data-driven techniques. As an example, Grubwinkler et al. present a statistical model derived from crowd-sourced power consumption data [7]. Concretely, their proceeding is depicted in Figure 2.4. First, they model the vehicle characteristics using a *Power Consumption Map* (PCM). The PCM provides the amount of energy consumed for each operating point, that is, for each speed and acceleration value. Second, they distinguish several road classes like highway or residential according to their definition on OpenStreetMap [13]. For each road type, they then derive a probability density function of operating points based on the crowd-sourced data. Given a specific road type, it describes how likely each operating point is. Together, this provides all the information needed to statistically predict the mean energy consumption for a certain route.
Figure 2.3: Comparing distance to empty estimated by Tseng et al. with Nissan Leaf range prediction. Source: Tseng et al. [20]

Figure 2.4: Statistical energy consumption model. Source: Grubwinkler et al. [7]
3 Problem Statement

In this paper, we take a data-driven approach, and investigate simple and practical models for range prediction for a given e-bike-rider pair. In the remainder of this paper, we discuss the challenges in identifying and extracting useful features from imprecise and often incomplete e-bike usage data, and we compare the accuracy of various models, from simple ones which only take prior riding behaviour into account, to more complex ones that also include route properties.

We consider two situations, as described in Table 3.1. In the first setting, we do not know the rider’s destination and therefore we can only extrapolate past energy consumption to predict the distance-to-empty. At a minimum, we need a time series of battery state-of-charge and odometer readings, from which we estimate the average battery consumption per kilometer. If we have corresponding measurements of tire pressure, battery temperature, or other parameters that may influence range, these can also be added. As a result, we will build one model per rider that reflects his or her typical battery consumption. The model can then be used for on-board range estimation by periodically probing the battery state-of-charge and estimating the number of kilometers that can be travelled.

Table 3.1: Two settings for range prediction studied in this work.

<table>
<thead>
<tr>
<th>Destination unknown</th>
<th>Destination known</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Goal</td>
</tr>
<tr>
<td>Distance-to-empty prediction</td>
<td>Predicting the battery charge at the destination</td>
</tr>
<tr>
<td>Prediction based on</td>
<td>Prediction based on</td>
</tr>
<tr>
<td>Past riding behaviour</td>
<td>Past riding behaviour; Anticipated route profile</td>
</tr>
</tbody>
</table>

In the second setting, the destination is also given, therefore we can add route-specific parameters to each rider’s model. This requires a richer dataset: in addition to a sequence of mileage and battery consumption values, we need timestamped GPS traces of the rider’s past trips, which we can cross-check against map and weather data to extract route-specific information that can influence range, including wind speed, change in elevation, the number of stop signs and the road surface (most e-bikes can be ridden on paved roads and off-road trails). These parameters can capture more information about riding habits such as how quickly a rider accelerates from a stop sign, and more information about the rider such as his or her weight (heavier riders will use more energy when going uphill). Then, when the rider specifies a destination, we can calculate the
shortest path to it and estimate the state-of-charge upon reaching the destination using the characteristics of the planned route. Of course, any data-driven model assumes that past history is a good predictor of the future. If a rider changes his or her riding style, our models must be recomputed.
4 Data processing and analysis

Before presenting our range estimation techniques, we discuss the challenges involved in obtaining the required data.

4.1 Data Sources

The main problem is that in contrast to cars, e-bikes do not come with an equivalent of an on-board diagnostics port and therefore there is no built-in way of collecting data. The battery is sealed and the digital display (recall Figure 1.1) has no data storage or transmission capabilities.

Fortunately, field trials are being conducted, in which participants ride e-bikes that are outfitted with sensors or smart phones [5, 17, 14, 22]. We obtained data from one of these field trials [22], taking place at the University of Waterloo, Canada, which includes measuring battery current and voltage. Data have been collected between April and October 2015 from 31 cyclists, chosen from interested graduate students, university staff and faculty members. Prospective riders filled out a questionnaire and the final 31 participants were selected based on their anticipated usage of the e-bike; roughly half the riders are male and half are female, and one-third each are students, staff and faculty.

Each participant was given an e-bike, shown in Figure 4.1a, equipped with a custom-made hardware kit shown in Figure 4.1b. The hardware kit is mounted on the battery and includes voltage, current and battery temperature sensors, as well as a smart phone. The phone battery is charged by the bike battery. To save power, the phone is in sleep mode for 58 seconds out of every minute, and takes four measurements during the two seconds it is active. The measurements include battery voltage, current and temperature, as well as GPS coordinates. When feasible, the collected data are uploaded to a server over WiFi.

In addition to obtaining the e-bike usage data, we created a routable street graph using OpenStreetMap [13]. The graph includes bike trails and footpaths as well as traffic lights and stop signs.

In the remainder of this chapter, we explain how to process and analyze the above data to obtain the variables we need for our range prediction models. Recall that if the destination is not given, we only require distance and battery consumption measurements. There is no direct access to the odometer on the bike, but we can reconstruct the routes travelled using GPS traces and map data (section 4.2). Furthermore, we can calculate battery consumption from the voltage and current measurements (section 4.4). Additionally, to take route properties into account, we need to look up the properties of the
CHAPTER 4. DATA PROCESSING AND ANALYSIS

(a) The e-bike used in the field trial. The motor is located at the bottom bracket, spinning the same shaft as the pedals.

(b) Hardware kit used to collect e-bike data. An aluminum box attached to the battery (upper-right corner) contains a smartphone and several sensors. An assembly guide can be found at [21].

Figure 4.1: The hardware setup of our e-bike field trial.

corresponding route segments in the OpenStreetMap data (section 4.3).

4.2 Map-Matching GPS Traces

The first data processing step is to reconstruct trip routes from GPS data. Recall that this is necessary to determine distance travelled and to link route-related features such as road surface or the number of stop signs with battery consumption. This task, known as map-matching GPS, amounts to matching GPS data points, referred to as GPS fixes, to an underlying road network.

There are many existing techniques that can solve this problem, from relatively simple to highly sophisticated ones; see [15] for a review. Most algorithms "are designed for use with high frequency positioning data (i.e. 1 Hz or 1 second sampling interval)" [16]. In addition, there are some approaches (e.g., [16], [24]) that specifically account for sampling intervals of 1 minute and higher. Since our data set contains bursts of up to four GPS readings every minute, we use an approach similar to that from Quddus et al. [16].

The main idea is to connect the observed GPS fixes via shortest paths consistent with the
vehicle’s trajectory. In addition, Quddus et al. take bearing information into account and introduce a weighing mechanism. We show that even without these advanced features, our algorithm provides results that meet our needs.

Algorithm 1 Matching GPS fixes with road graph edges

1: function \texttt{MapMatch}(G, [f_1, f_2, \ldots, f_n]) \quad \triangleright \quad n \geq 2
2: \quad u \leftarrow \texttt{GetClosestEdge}(f_1, G) \quad \triangleright \text{Start edge}
3: \quad s \leftarrow \texttt{GetClosestNode}(f_1, u) \quad \triangleright \text{Start node}
4: \quad M \leftarrow \texttt{[]} \quad \triangleright \text{Result list}
5: \quad i \leftarrow 2
6: \quad \textbf{while} \ i \leq n \ \textbf{do}
7: \quad \quad v \leftarrow \texttt{GetClosestEdge}(f_i, G) \quad \triangleright \text{Target edge}
8: \quad \quad t \leftarrow \texttt{GetClosestNode}(s, v) \quad \triangleright \text{Target node}
9: \quad \quad M.\texttt{append}((\texttt{GetShortestPath}(s, t, G)))
10: \quad \quad \quad i \leftarrow i + 1
11: \quad \quad \quad s \leftarrow t \quad \triangleright \text{Shift start node}
12: \quad \textbf{end while}
13: \quad \textbf{return} \ M
14: \textbf{end function}

Algorithm 1 shows the details. As input, we take in a routable road network graph \(G\) and a sequence of GPS fixes \([f_1, f_2, \ldots, f_n]\). The output is a list of matched road graph edges \(M\) that represent the reconstructed route. The algorithm is divided into an initialization part (lines 2-5) and a loop (lines 6-12). The initialization part identifies a node \(s\) from the road graph \(G\) representing the trip’s start position. This can be achieved by first searching for the edge \(u\) that is closest to \(f_1\) (line 2). We define the distance between a node and an edge in a standard geometric sense, as the length of the shortest straight line connecting both. As an optimization, we may only consider edges within a certain radius. Second, we look at the two nodes defining \(u\). We select the one that is closer to \(f_1\) and call it our start node \(s\) (line 3).

Next, the algorithm iteratively considers the section from the current to the next GPS fix and then moves on to the next pair of fixes. Let \(f_i\) be the current fix and \(s\) the (recently found) corresponding node of the road graph. Similarly to the initialization step, we first search for the edge \(v\) that is closest to \(f_{i+1}\). Second, we compare the two nodes defining \(v\), select the one that is closer to \(s\) and call it \(t\). Third, we append all edges that belong to the shortest path between \(s\) and \(t\) to our result list \(M\). Finally, we let \(t\) be the new start node \(s\) and move on to the next fix.

Figure 4.2 shows the results of the algorithm on three GPS traces from three different e-bikes. The red dots indicate the GPS fixes. Note the shortest-path interpolation in Trip C, which has missing GPS data in the middle.
Figure 4.2: Evaluating the map-matching algorithm on three sample trips.
4.3 Trip Segmentation

The next step is to divide a trip into segments, i.e., consecutive road graph edges. We will then build a model that correlates battery consumption with various properties of a segment (length, number of stop signs, etc.). In order to be included in the set of segments that will be used to build the model, a segment must satisfy the following three properties:

**Minimum segment length** \( \alpha \) We denote the minimum length of a trip segment by \( \alpha \). We may need to combine multiple consecutive road graph edges until their total length is at least \( \alpha \).

**Minimum riding duration** \( \beta \) We denote the minimum time of travel for a segment by \( \beta \). This is important to ensure an accurate speed calculation, i.e., combined segment lengths divided by combined riding duration.

**Maximum GPS time gap** \( \gamma \) The maximum duration for which GPS may be unavailable is denoted \( \gamma \). If GPS data are missing for a long period of time, the shortest path assumed by Algorithm 1 may not necessarily be the path taken by the e-bike, so we want to exclude such trip sections.

Since the data set we are using contains bursts of measurements every minute, we set \( \beta = 60 \text{ s} \). We tested different values of \( \gamma \in [80 \text{ s}, 300 \text{ s}] \) and found that the total number of road segments we identified did not change significantly, so we conservatively set \( \gamma = 80 \text{ s} \). Increasing \( \alpha \) leads to a decreasing total number of obtained trip segments but on the other hand results in higher variance of energy consumption per segment. As a trade-off, we select \( \alpha = 400 \text{ m} \).

Given these parameters, the trip segmentation algorithm produced 3619 segments from the field trial dataset. Figure 4.3 shows a segmentation of a sample trip; just for this illustration, we increase \( \alpha \) to 700 m to make the segments more visible. The blue, red and pink lines represent the created segments. They all have approximately equal length. In between the blue and red segments, no GPS data were logged, probably due to missing satellite connection. As the duration of GPS being unavailable exceeds \( \gamma \), this section is not included in our set of 3619 trip segments.

Finally, for each segment, we obtain its properties from OpenStreetMap data, such as road surface, the number of stop signs, etc.

4.4 Obtaining and Analyzing Energy Consumption

The last piece of the puzzle is to estimate the battery consumption for each trip segment, call it \( \Delta E \). We do this by linearly interpolating the product of voltage and current over time. Let \( t_{\text{start}} \) and \( t_{\text{end}} \) be the times when the rider entered and left the trip segment, respectively. Let \( U(t) \) and \( I(t) \) denote voltage and current at time \( t \), respectively. Moreover, let \( t_0, t_1, \ldots, t_k \) be the times at which voltage and current were measured.
Note that our trip segmentation algorithm ensures that $t_{\text{start}} = t_0$ and $t_{\text{end}} = t_k$. We approximate
\[
\Delta E = \int_{t_{\text{start}}}^{t_{\text{end}}} U(t)I(t)dt
\]
by computing the Riemann sum
\[
\sum_{i=1}^{k} U(t_i)I(t_i)(t_i - t_{i-1}).
\]
Finally, we divide $\Delta E$ by the length of the segment to obtain a battery consumption value in units of Wh/km.

Figure 4.4 shows a histogram of the distribution of energy consumption for all 3619 trip segments. We observe a broad range of per-segment consumption values, most of which range between zero and 30 Wh/km. The spike at 0 Wh/km corresponds to trip segments where riders did not use the motor. This happens after the riders have reached their desired speed and are coasting, usually downhill—the motor always spins whenever the rider is pedalling—or after they have reached the e-bike’s maximum speed of 32 km/h and the motor disengages automatically. The per-segment average works out to 7.9 Wh/km. The e-bikes from which we obtained data have battery capacity of 330 Wh, which corresponds to an average range of $\frac{330}{7.9}$, or about 42 km. This is quite close to the manufacturer-specified maximum range of 45 km [6].

Next, we analyze the energy consumption of different riders. Of the 31 total riders, only 13 had at least 80 road segments. Figure 4.5 shows a clustering of the average energy consumption of these 13 riders: economical riders consuming 5 to 6 Wh/km, average riders consuming around 8 Wh/km, and aggressive riders consuming over 9 Wh/km.
4.4. OBTAINING AND ANALYZING ENERGY CONSUMPTION

Note that the rider with the highest consumption (9.9 Wh) uses, on average, more than twice as much energy per kilometre as the most economical rider (4.8 Wh). This motivates the need to develop per-rider range prediction models rather than using the manufacturer-specified maximum range for everyone.

Finally, in Figure 4.6, Figure 4.7 and Figure 4.8 we show the consumption histograms for the 13 riders with at least 80 road segments, split by cluster. Each histogram shows noticeable spread, meaning that the same rider may use different amounts of energy in various trip segments.

Figure 4.4: Energy consumption distribution for the 3619 trip segments identified from the data.

Figure 4.5: Clustering riders by normalized mean energy consumption
CHAPTER 4. DATA PROCESSING AND ANALYSIS

(a) Mean: 4.8 Wh/km
(b) Mean: 6.4 Wh/km
(c) Mean: 5.7 Wh/km
(d) Mean: 6.4 Wh/km

Figure 4.6: Energy consumption distributions: Cluster of "economical" riders

(a) Mean: 7.8 Wh/km
(b) Mean: 7.4 Wh/km
(c) Mean: 8.3 Wh/km
(d) Mean: 7.7 Wh/km
(e) Mean: 8.4 Wh/km

Figure 4.7: Energy consumption distributions: Cluster of "average" riders

(a) Mean: 9.5 Wh/km
(b) Mean: 9.3 Wh/km
(c) Mean: 9.3 Wh/km
(d) Mean: 9.9 Wh/km

Figure 4.8: Energy consumption distributions: Cluster of "aggressive" riders
5 Range Prediction

We now evaluate two methods for range prediction corresponding to the two cases mentioned in chapter 3. For situations where the destination is not given, we study mean prediction, i.e., we use the average historical consumption (recall Figure 4.5) of the rider to estimate his or her range. If the destination is given, we consider a linear regression model that uses route-specific details. We reiterate that to account for individual riding behaviour, we fit a separate model for each rider.

We use the first 120 km of each rider’s trips as the training data and we use the remaining data to test the models. To prevent overfitting, in this chapter we only use the top four most frequent riders, summarized in Table 5.1. The other riders have fewer than 300 trip segments and not much more than 100 km of riding data. To evaluate prediction performance, we examine how much the observed consumption at the end of a trip in the testing part of the data set differs from the predicted consumption. We normalize these "per-trip" residuals by dividing them by battery capacity because our ultimate goal is to determine the “distance-to-empty” which depends on the battery size, rather than an individual trip.

Table 5.1: Characterizing the four riders used for model fitting

<table>
<thead>
<tr>
<th>Rider</th>
<th>Mean consumption</th>
<th>Average trip length</th>
<th>Number of trip segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.7 Wh/km</td>
<td>3.5 km</td>
<td>317</td>
</tr>
<tr>
<td>2</td>
<td>8.3 Wh/km</td>
<td>2.8 km</td>
<td>752</td>
</tr>
<tr>
<td>3</td>
<td>8.4 Wh/km</td>
<td>6.3 km</td>
<td>306</td>
</tr>
<tr>
<td>4</td>
<td>9.9 Wh/km</td>
<td>4.6 km</td>
<td>653</td>
</tr>
</tbody>
</table>

5.1 Mean Prediction

The mean prediction model is simple to compute and easy to implement in practice on an e-bike display. It suffices to periodically record the battery voltage and current, and the odometer reading\(^1\) and maintain the mean consumption. Figure 5.1a shows the distribution of the per-trip normalized residuals. For 89 percent of trips, the predicted consumption deviates from actual consumption by at most ten

\(^1\)Odometer data is already available to the e-bike display unit. As we explained earlier, we use GPS data instead of an odometer because this data was not exported out of the unit.
percent of the battery capacity. Depending on the rider, the normalized root mean squared errors (RMSE) range from 4.9 to 7.4 percent, as listed in Table 5.2. Plotting the normalized residuals versus the corresponding test trip lengths (see Figure 5.1b) reveals that short and long trips appear to be predicted equally well (and that there are more short trips under 4km than long trips).

Table 5.2: Per-trip root mean squared errors of mean prediction

<table>
<thead>
<tr>
<th>Rider</th>
<th>RMSE</th>
<th>RMSE normalized by battery capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17.4 Wh</td>
<td>5.3%</td>
</tr>
<tr>
<td>2</td>
<td>16.2 Wh</td>
<td>4.9%</td>
</tr>
<tr>
<td>3</td>
<td>24.3 Wh</td>
<td>7.4%</td>
</tr>
<tr>
<td>4</td>
<td>20.6 Wh</td>
<td>6.2%</td>
</tr>
</tbody>
</table>

The accuracy of the simple mean prediction technique may seem surprising given what we have seen in Figure 4.6, Figure 4.7 and Figure 4.8. This may be explained as follows. Clearly, using mean prediction to estimate battery consumption for a single trip segment is inaccurate because the same rider may use a different amount of energy on different trip segments. However, when predicting the remaining range at the end of a trip, as we do here, some segments are underestimated while others are overestimated, and the per-segment errors roughly cancel each other out.
5.2 Linear Regression

5.2.1 Model Definition

We now investigate the predictive power of incorporating all available data into the model. To predict the energy consumption for a given trip, we first split it into segments as described in section 4.3. For each road segment, we model the energy consumption $\hat{y}$ as a linear function of four variables: the average speed $v$, the number of traffic lights, stop signs and give-way signs $z$, the off-road fraction of the segment (i.e., foot paths and bike paths where cars are not allowed) $r$, and the average battery temperature $\theta$:

$$\hat{y} := m_v v + m_z z + m_r r + m_\theta \theta + b.$$ 

For each rider, we then use ordinary least-squares linear regression to obtain the coefficients $m_i$ for $i \in \{v, z, r, \theta\}$. $b$ is the intercept term. Table 5.3 describes the features of the model in more detail. We calculate average speed by dividing the length of a segment by the time taken to ride it. We obtain $z$ and $r$ from OpenStreetMap, and we get $\theta$ from the temperature sensor that is included in the hardware kit. Note that our dataset does not include any information about the riders themselves (such as weight) or any weather information (such as wind or precipitation). Also, even though the change in elevation throughout each trip segment can be computed based on satellite raster images (available from e.g., [9]), the topology of Waterloo (the city where the e-bike trial is taking place) is relatively flat, and so most of the trip segments we identified have little or no change in elevation. Thus, we do not use this parameter in our regression model.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average speed</td>
<td>$v$</td>
<td>Riding duration divided by segment length</td>
</tr>
<tr>
<td>Traffic signs</td>
<td>$z$</td>
<td>Number of traffic lights, give-way and stop signs normalized by segment length</td>
</tr>
<tr>
<td>Off-road fraction</td>
<td>$r$</td>
<td>Covered off-road riding distance divided by segment length</td>
</tr>
<tr>
<td>Average battery temperature</td>
<td>$\theta$</td>
<td>Average battery temperature</td>
</tr>
</tbody>
</table>

The linear regression model is somewhat more difficult to implement in practice. We require a navigation-like interface on the e-bike to input the planned destination, and a copy of a routable road graph to determine the shortest path to the destination, and to extract the features we need for each segment of the anticipated route. To determine whether there is sufficient range to reach the destination, we then sum up the predicted
battery consumption for each segment of the planned route. We also need to train the regression model using past history, and perhaps periodically re-compute the model as more data become available. This can be done aboard the bike or remotely at the server that collects the data. For the training dataset, we need battery voltage and current sensors, as before, plus GPS data and battery temperature.

A subtle issue arises when predicting the average speed $v$ for each road segment along the planned route. In the training data set, the average speed can easily be calculated by dividing the length of a trip segment by the time it took the rider to travel through it. However, we do not know what the speed will be on the planned route. In our analysis, we can “cheat” since even the trips on which we are testing the model have already taken place. Thus, for this analysis, we input the actual average speed per segment to our model. This way we can directly examine the correlation of observed speed and energy consumption. However, in practice, we need a way to predict what the speed is going to be. This can be done by examining this rider’s average speed on (similar road segments in) previous trips, or by searching for similar riders who have already travelled these particular road segments, similar to the crowd-sourcing approach of Tseng et al. [20] that we mentioned in chapter 2.

5.2.2 Results

Before fitting the regression model, we analyze the distribution histograms of the four features in Figure 5.2 to ensure that the data look “reasonable” and there are no obvious data quality problems; these histograms include data from all four riders. First, note that some road segments are showing an unusually high average speed of 45 km/h or more. There are at least two possible explanations. One is that some riders have taken their e-bikes on a bus or a train for some portions of their trips. Another possibility is that the OpenStreetMap data do not include all possible routes that an e-bike could take (e.g., there may be unknown shortcuts through a park or a playing field). Figure 5.3 illustrates this issue based on the example of a construction site. In the upper right corner, the satellite image reveals missing connectivity of the OpenStreetMap road network. Consequently, the map-matching algorithm (recall section 4.2) cannot do better than outputting a long roundabout route. This results in a calculated average speed of 79.6 km/h which is obviously unrealistic. As a result, we remove all trip segments with an average speed above 45 km/h.

As for the other features, the number of traffic lights and stop signs is often zero and rarely more than four, the average battery temperature is around 25 degrees Celsius (most of our dataset was collected in the spring and summer), and the off-road percentage is usually zero (e.g., a regular city street) or 100 percent (e.g., a dedicated bike path).

Now, after using the ordinary least squares method to fit four regression models, one for each rider, we obtain a coefficient of determination $R^2$ of zero. By definition, this means the estimator does not perform better than one that simply predicts the overall mean consumption across all riders. Accordingly, the residual plots (Figure 5.4) resemble
those obtained by mean prediction (Figure 5.1). Table 5.4 lists the per-trip prediction errors for each rider. They turn out to be slightly higher than the mean prediction errors (see Table 5.2) which clearly indicates overfitting.

To check if the features correlate with energy consumption in a non-linear fashion, we perform residual analysis. For each feature, we plot a subset of the residuals where the other features are approximately constant. For example, when plotting residuals versus \( r \), we only consider trip segments with

\[
    z = 0 \\
    \text{and } |v - \overline{v}| < \frac{5 \text{ km}}{\text{h}} \\
    \text{and } |\theta - \overline{\theta}| < 5 ^\circ \text{C},
\]

where \( \overline{v} \) and \( \overline{\theta} \) denote average speed and temperature, respectively. Figure 5.5 shows the (normalized) residual plots. We see no obvious evidence of non-linearity. Thus, it appears that although the features we have chosen seem reasonable, they do not improve range prediction beyond the simple mean prediction model.

5.2. LINEAR REGRESSION

Figure 5.2: Distributions of average speed, number of traffic signals, average battery temperature and off-road percentage.
CHAPTER 5. RANGE PREDICTION

Figure 5.3: Map-matching failure due to shortcomings in the underlying data

Table 5.4: Per-trip root mean squared errors for linear regression

<table>
<thead>
<tr>
<th>Rider</th>
<th>RMSE</th>
<th>RMSE normalized by battery capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17.5 Wh</td>
<td>5.3%</td>
</tr>
<tr>
<td>2</td>
<td>18.7 Wh</td>
<td>5.7%</td>
</tr>
<tr>
<td>3</td>
<td>31.8 Wh</td>
<td>9.7%</td>
</tr>
<tr>
<td>4</td>
<td>20.8 Wh</td>
<td>6.3%</td>
</tr>
</tbody>
</table>

Figure 5.4: Evaluating the linear regression approach

(a) Distribution of residuals per test trip

(b) Residuals versus trip lengths
5.2. LINEAR REGRESSION

Figure 5.5: Normalized residual plots for each feature.
6 Conclusions and Future Work

In this paper, we initiated the study of range prediction for electric bicycles. Using real e-bike usage data, we analyzed two techniques: one that only uses average battery consumption from past trips, and one that uses a linear regression model with route-specific information such as the number of stop signs and traffic lights. Our main findings are as follows:

1. As we showed in Figure 4.4 and Figure 4.5, riding behaviour matters. The most economical rider in our dataset was over twice as efficient, on average, as the most aggressive one. Thus, the manufacturer-specified maximum range is not an accurate predictor.

2. Even though a particular rider may use a different amount of energy under different circumstances (Figure 4.6, Figure 4.7 and Figure 4.8), we found that in flat regions, range prediction based on historical mean energy consumption works well at the granularity of trips. Using real trip data, our prediction errors at the end of a trip were usually within ten percent of the total battery capacity.

3. Surprisingly, we found that a linear regression model using average speed per route segment, the number of traffic signs, off-road percentage and battery temperature performed no better than mean prediction.

4. To save power, e-bike sensor data may need to be collected with lower frequency than desired. As we discussed in section 4.2, map-matching GPS data generated by e-bike usage required an algorithm that can handle sparse data. Moreover, as we mentioned in subsection 5.2.2, e-bikes (and bikes in general) are not restricted to travelling on city streets and well-marked paths, meaning that some errors in interpolating routes between GPS fixes are difficult to avoid.

Our results are of interest to e-bike manufacturers. Our findings suggest that a simple on-board prediction technique can be implemented by measuring battery voltage, battery current and mileage. Most e-bikes already come with an odometer, so by making odometer data accessible, and deploying additional sensors at the battery, our simple technique can be implemented inexpensively. Since range anxiety is a major barrier to the adoption of electric vehicles, adding such a range estimator can help widen the appeal and increase the sales of e-bikes.

We hope that this paper encourages further work in this area. One critical direction for future work is to validate our findings on other, perhaps larger, data sets (however, to
the best of our knowledge, the data set analyzed in this paper is the largest currently available e-bike usage data set. It would also be useful to investigate whether adding more variables to the regression model, such as wind speed or change in elevation, can improve performance over the simple mean prediction model.
Bibliography


[21] University of Waterloo ISS4E group. Assembly guide for the WeBike hardware kit. URL: http://blizzard.cs.uwaterloo.ca/iss4e/?page_id=4945 (visited on 01/07/2016).

[22] University of Waterloo ISS4E group. WeBike project homepage. URL: http://blizzard.cs.uwaterloo.ca/iss4e/?page_id=3661 (visited on 12/10/2015).
