A survey of evaluation methods for personal route and destination prediction from mobility traces

Roelant A. Stegmann[*], Indrė Žliobaitė[†*], Tuukka Tolvanen[*], Jaakko Hollmén[*], Jesse Read[‡]

Article Type:

Overview

Abstract

Personal mobility data can nowadays be easily collected by personal mobile phones and used for analytical modelling. To assist in such an analysis, a variety of computational approaches have been developed. The goal is to extract mobility patterns in order to provide traveling assistance, information, recommendations or on-demand services, for instance. While various computational techniques are being developed, research literature on destination and route prediction lacks consistency in evaluation methods for such approaches. This study presents a review and categorisation of evaluation criteria and terminology used in assessing the performance of such methods. The review is complemented by experimental analysis of selected evaluation criteria, to highlight the nuances between the evaluation measures. The experimental study is using previously unpublished mobility data of 15 users collected over a period of six months in Helsinki metropolitan area in Finland. The paper is primarily intended for researchers developing approaches for personalised mobility analysis, as well as a guideline for practitioners to select criteria when assessing and selecting between computational approaches. Our main recommendation is to consider user-specific accuracy measures in addition to averaged aggregates, as well as to take into consideration that for many users accuracy does not saturate fast and the performance keeps evolving over time. Therefore, we recommend using time-weighted measures.

[*]Department of Computer Science, Aalto University, Finland

[†]Department of Computer Science; Department of Geosciences and Geography, University of Helsinki, Finland

[‡]Laboratoire d'Informatique (LIX), École Polytechnique, France

1 Introduction

Data on personal mobility is increasingly easy to collect, as location recording devices are widely available. Nowadays every (smart) mobile device incorporates a GPS rceiver, able to track location accurately. Along with advancing technology, research interest in analysing mobility data has been rapidly increasing, as illustrated, for instance, by the fact that Google Scholar has four times more papers mentioning "GPS traces" in 2014-2015 compared to 2000-2001.

One major line of data science research focused on location data aims to predict the destination (where the user is going) and route (how the user is going there). Destination prediction and route prediction are the focus areas of this survey paper. Here the primary target is to predict geographical locations in spatial domain as accurately as possible, with travel times as a secondary target. These prediction tasks are motivated by at least four types of application needs. Firstly, there is a demand to provide personalised services. Many phone applications already provide their users with information about, for instance, the travel time to home at the end of a working day, route conditions, or traffic jams, for instance. Secondly, predictions may help to save energy on the societal level, for instance, by fuel savings via carpooling or real time route optimisation [24, 18, 67, 21]. Thirdly, predictions may be used for providing personalized content, for instance, via location-based social networks [15]. Finally, aggregated predictions can be used for traffic management and long-term strategic planning, for instance, optimising night bus routes by taxi traces [11], predicting taxi demand [52], or validating bus schedules [50].

This survey has two objectives: on the one hand, to present a systematic overview of research task settings for predictive modelling of personal mobility patterns, and on the other hand, to define performance evaluation measures for different prediction scenarios that can be used by researchers in the field. At the moment, consistency is lacking which makes it difficult to compare different methods and systems—and thus to show relative improvements of novel, proposed solutions. By providing a systematic evaluation methodology and a unified terminology, we hope to provide a frame of reference for researchers and practitioners in this rapidly growing research area.

2 Characteristics of mobility data

We start with a discussion on the basic characteristics of data used in mobility analysis, route and destination prediction, and the terminology used in this research domain, to provide a unified frame of reference for researchers.

2.1 Scenarios for predictive modeling

Consider two scenarios as examples to establish a notation for the location data and the parameters used in the paper. In scenario one, a person is moving with his personal smartphone, which collects location traces. Several applications tap into the destination and route prediction build on top. When he opens the travel planner app, it suggests the four most likely next destinations for quick access. A car sharing app uses route prediction to enable

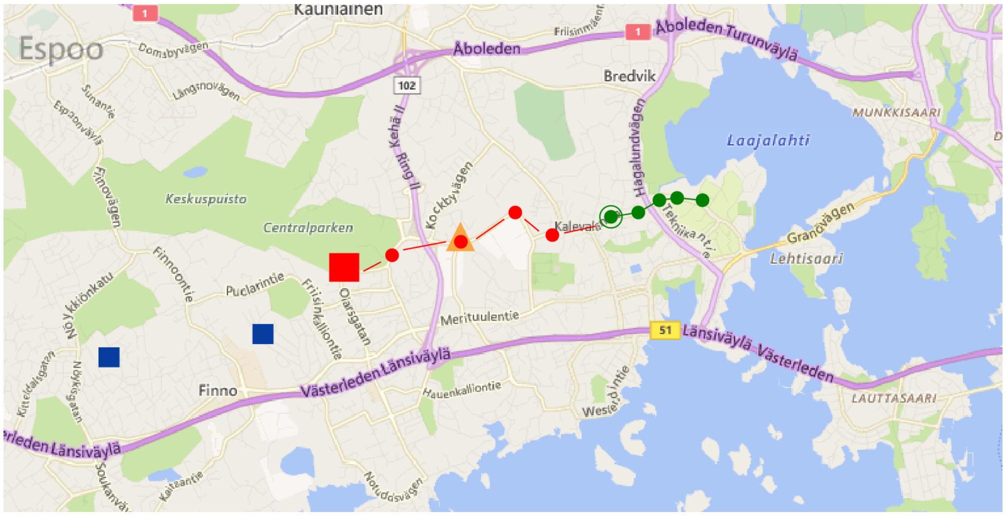

Figure 1: The thick green connected dots in the centre are the recent trace, with the last (and current) location as the left-most green point. Destination identification is identifying the locations (blue squares). Destination prediction is predicting the next-up destination (large red square). Location prediction is predicting a location a certain time ahead (orange triangle). Route prediction is predicting the route given the next destination (red connected dots).

others to ask for a hypothetical ride; only if someone is interested in the predicted route, they confirm whether the user is actually travelling it. In scenario two, taxis constantly report their location to the control room. In order to allocate taxis efficiently, we do not only need to know where taxis are and where taxis with a ride are going; but also want to predict where empty taxis (given their location now) will be in e.g. thirty minutes—location prediction. See [52] for an example of the latter scenario.

Figure 1 illustrates different scenarios for predictive modeling in personal mobility analysis. Destination prediction is predicting the next-up destination. Location prediction is predicting a location a certain time ahead. Route prediction is predicting the route given the next destination.

2.2 Concepts and notation

In both of the scenarios mentioned earlier, a GPS device e tracks its own location l^e over time $l^e(t)$ in coordinates $l^e = (lon, lat)$ in the spatial domain. Such a device is generally associated to a vehicle v or a user u. Any mapping which is not one-to-one such as multiple users of one device or multiple devices for one user needs to be pre-processed carefully and will complicate the research. In this paper, we assume the data is mapped to a single user, and use the notation u throughout. The superscript is sometimes dropped when it is clear from the context.

The unique location for a user u over time $l^u(t)$ is only available at the times t it has been sampled. All these times t are denoted by the (ordered) set T, which spans a certain time interval. Of course, we will work with the assumption of sampled data with a corresponding,

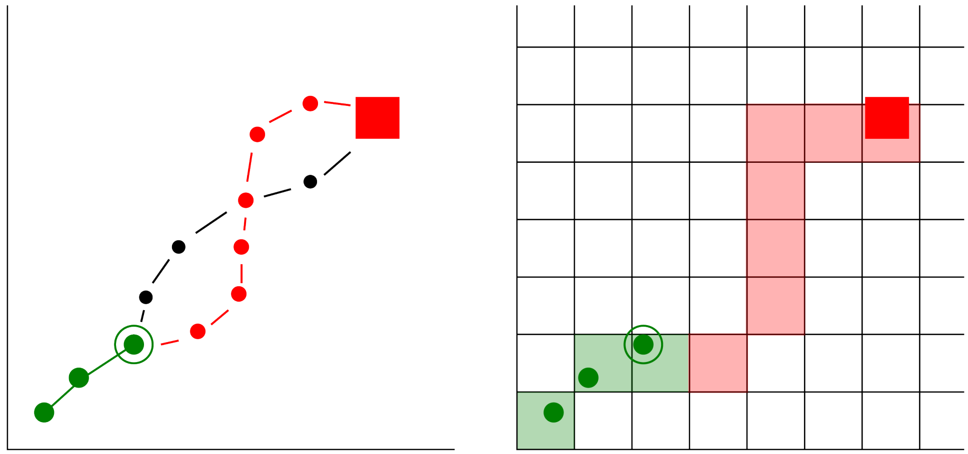

Figure 2: Illustration of network (left) and grid (right) structure. Note that several points might snap to one grid, and that this specific grid drawing allows for diagonal transitions.

finite sampled period. Generally the points in T have regular intervals $\Delta t = \Delta(t_1 - t_2)$. The work presented in [45] illustrates how predictability of location is still possible, despite of the very large sampling intervals. If not, the data is often pre-processed to obtain regular interval samples and make it more suitable for machine learning techniques—earlier work of the authors focused on the challenges of this pre-processing [73, 60]. While data-preprocessing and filtering are out of the focus of this study, an interested reader is referred to [14] for a comprehensive overview.

A GPS location trace Tr of length N is a sequence of (GPS) locations from

$$Tr^u = \big(l^u(t_1), l^u(t_2) \dots, l^u(t_N)\big). \tag{1}$$

We will assume this data is put in a vector and simplify the notation to

$$Tr^u = \big(l_1^u, l_2^u, \dots, l_N^u\big). \tag{2}$$

Of course, other representations than (lon, lat)-coordinates can be used for the location. A lot of research use waypoint representation [77, 64], where every location is a waypoint w, a member of pre-defined locations. One approach is using road intersections as waypoints, creating a directed graph representation of the road network. Each GPS point is then snapped (or quantized) to the nearest waypoint. Another approach is to impose a grid over the traces [14, 76] and snap the points to its containing cell—a grid is not necessarily square but could also be hexagonical [3]. This creates a grid structure, which can be considered a network in which each cell is connected to its direct neighbours. See Figure 2 for an illustration of these two structures. The grid structure requires a higher sampling frequency, but no predefined road network which in turn the waypoint representation requires. Related work considers constructing such a road network from traces themselves [10, 46].

Humans typically generate rather regular travel patterns, with destinations that are visited periodically, such as home, university, work, and a limited set of shops, events, and meeting places, for instance. An overview of activities and travel duration and their

Table 1: Summary of notation user in the paper

Symbol	Definition
$l^e(t)$	**location** over time t on device e
$Tr^u = (l^u_1, \ldots, l^u_N)$	A **trace** (of user u, length N)
u, v, e	**user**, vehicle, device
$t \in T$	'set' of times
$(\hat{d})$ d	(predicted) destination
$D^u = \{d^u_k\}_{k=1}^{\lvert D^u \rvert}$	Set of **destinations** of user u
$k = \mathsf{rank}_{D^u}(d^u)$	k-th most important destination
r	range (meters) within which a user stays
b	minimum time (seconds) within range r to be a destination
$\mathsf{dist}(\cdot, \cdot)$	distance function
X	set of (mapped) traces
$R^u = (l_1, \ldots, l_N)$	a **route** where $l_1 \in D^u$ and $l_N \in D^u$
$S = \{R_1, \ldots, R_N\}$	set of routes
$(\hat{l}_{t+x})$ l_t	(predicted) location at time t (x time from now)
$(\hat{y}_i)$ y_i	(predicted) i-th destination/location/route
$(\hat{Y})$ Y	set of (predicted) locations or destinations, $\lvert Y \rvert = N$
w	waypoint

motivations, for a group of users is given in [42, 39]. For a person or user u these can be several destinations d^u or points of interest, sometimes called *target* or *goal*, but these are rather unspecific terms. Any d^u is a location and could be said to be associated with a certain rank $k = \mathsf{rank}_{D^u}(d^u)$ among all other destinations D^u of the user. Typically, destinations are identified in the data by defining a threshold b (break) for the time that the user stays within range r—respectively in seconds and in meters throughout this work. The system then keeps a count for how often a user is in each destination. Clustering methods are also used to identify often visited locations, for instance via DBSCAN clustering algorithm [33, 34].

It helps to use the word *trace* (as in, *GPS trace*) for raw data, *trajectory* for trace data mapped to a grid or graph representation, and *route* or *trip* for a trajectory from a start point to an end point. A route R of length N is a sequence of locations l where one generally assumes that the first and last are also a destination d^u:

$$R^u = (l_1, l_2, \ldots, l_N). \tag{3}$$

For clarity, we present a summary of the notation used throughout this paper in Table 1.

2.3 From raw location data to features describing mobility

Sampling interval or granularity of the data plays an important role in mobility data analysis, because traces with regular intervals are much easier to use than with irregular intervals. If presented with a choice, one would generally aim for smaller intervals for research purposes, because they allow more precise treatment. The clear trade-off is with battery power and network load, which is especially important when the data is collected on personal mobile devices and trasferred to a remote server.

From the raw loctaion data, higher-level features can be constructed that often aid the prediction. The most obvious and common features are speed, direction, and acceleration, which can be calculated easily from a set of time-indexed points. The main idea is to describe mobility patterns in a way that is invariant to data collection granularity, phase or length of the trip. Challenges of data pre-processing and feature construction are covered in [73, 60] in more detail. Visual analysis of mobility patterns is essential for identifying potential features and patterns. A detailed overview of visual analysis techniques in relation to mobility data can be found in [4].

Often, in addition to features describing mobility traces, contextual data describing the environment is also considered. Several papers investigate how contextual information improves the performance of predictive models [40, 49, 3]. The most common examples are time-context such as the day of the week, or part of the day (morning/evening) [28]. Weather information is sometimes also incorporated [66]. These are all contextual information as a function of time. Functions of the spatial location and domain could also be derived, such as the distance to the nearest bus-stop. Activity recognition can also be considered contextual information, where the activity is recognised by the use of for example GPS and often additional sensors such as the accelerometer. Even user emotion is sometimes added as context [19]. A good overview of contextual information is provided in [1] and more specific for mobile-computing in [12].

2.4 Privacy considerations

Gathering and analysing such mobility data undoubtedly raises privacy concerns. The ability to predict movements of individuals [65] makes it possible to identify individuals in combined GPS location data [17]. A good starting point of techniques and considerations can be found in [36].

The first issue to consider is whether the data can be linked to individuals. Tracking public transport routes, for instance, have little privacy issues because many individuals may be present at the same bus. If data can be linked to individuals, at least two types of methods can be used to guarantee privacy: data separation or density based methods. Data separation requires models to be made for each user individually using only their data. The disadvantage of this approach is that the performance of a model for a new user is likely to be worse since no data is yet available, and similarly for irregular travel patterns which have not yet shown in the data and thus are difficult to predict. For example, a user might not have yet been to the airport so it will be impossible to predict this location.

The second approach is density based, upon which only destinations and routes shared by several individuals will be used to strengthen the models of other users—see for example [70]. Although this has clear performance potential, its parameters need to be carefully chosen and tested to guarantee the privacy of users. As such, it can only work on rather large user bases. Alternatively, constraints may be imposed on data collection, for instance, by imposing a minimum number of locations a user has to have visited before his or her data can be shared into a common pool of other users [76]. For examples of techniques used to guarantee individuals' privacy see [30, 63].

3 Machine learning techniques for mobility prediction

Next we present an overview machine learning techniques that are commonly used in mobility analysis. It is not within the scope of this paper to provide an extensive technical review of machine learning techniques used, but to provide an overview of what types of techniques are commonly used. Table 2 presents a summary of scenarios, techniques and data characteristics reported in research literature. Studies included are typically focused on one or a few cities where traffic tracking data is accessible - e.g. a lot of studies consider Beijing. The road network mapping explained before is used more often than the grid mapping, probably because it preserves the intuitive graph structure of the data. Various computational algorithms are used, to be overviewed next. Accuracy measures vary across the papers, organising and interpreting these measures is the main subject of our survey.

3.1 Hidden Markov models

A Hidden Markov Model (HMM) is a probabilistic model that can be visually presented as a graph in the framework of graphical models [6]. The building blocks of HMMs are the first-order Markov chain to govern the temporal development of the hidden state information. The temporal dynamics of the discrete state information is modeled by the transition parameters between unobserved (hidden) states in time. The probability parameters between the current state and the next state form a transition matrix. In each state, there is a probability distribution of generating an observation.

The Markov models find their uses in for example Google PageRank and audio-to-text recognition. For a general review of HMMs, see [80] or, for modified architectures, see [7].

In our case, the nodes would be the waypoints (in the network representation) or cells (grid representation) and the transitions probabilities specify the likelihood of that next turn. The main advantage of HMMs is their simplicity and intuitive implementation in this situation. They work also with discrete output spaces, in contrast to some related Bayesian filtering methods (e.g. Kalman filters) which are suited to predict into real-valued space [61].

3.2 String matching

Generally speaking, strings are often used as a lower dimensional representation of sequences (e.g. Symbolic Aggregate Approximation (SAX) [44]). This representation has as an additional advantage that one can use many efficient string-matching algorithms that have been developed. As an illustration, a route might be a sequence of turn information LSSSRLSSSS (L: left, S: straight, R: Right). The main disadvantage is the high dependency on the initial starting point. This can be solved by assigning symbols to each waypoint - this allows for partial string matching algorithms to match routes which at least partially overlap. For an example using cellular data, see [40].

3.3 Temporally-augmented predictive models

A temporally-augmented predictive model stacks data instances across time, i.e., creates a window of examples as input, to predict a destination. Thus, it is straightforward to cast many mobility problems as a standard dataset suitable for the application of off-the-shelf supervised machine learning methods. methods, and in particular Random Forest (RF) have been shown to deliver very good results across several areas, and as such have also been implemented to this area. An advantage is that it allows for an easy integration of the aforementioned contextual information and features, and are relatively quick to train. This kind of approach was taken in [72] with, among other methods, ensembles of incremental decision trees.

3.4 Autoregressive time series models

Autoregressive time series models like autoregressive integrated moving average (ARIMA) models [8] are well-suited to time dependent signals, especially when seasonality and long term trends are present. However, this approach is thus principally suitable for continuous data, such as taxi *demand* prediction [53], rather than *mobility* (location and destination) prediction as considered by this work, which involve less pronounced trends and largely discrete-value prediction.

3.5 Recurrent neural networks

A recurrent neural network (RNN) is a neural network that incorporates the temporal structure of the data in internal, recurrent, connections (rather than a sliding window over past instances, as in the temporally-augmented models). The networks are notoriously difficult to train, but have drawn a lot of attention recently with the uptake of Long Short Term Memory (LSTM) networks [29], deep learning and graphic processing unit (GPU) boosted neural nets. For a review of RNNs and their variations, see [20].

3.6 Instance-based prediction

The main principle behind all the above described techniques (Markov models, temporally-augmented predictive models, string matching, and and recurrent neural networks) is to decompose possible route network into modules (e.g. based on crossings) and predict the next module from the most recently observed modules, with or without taking into account contextual information, such as time of the day or day of the week. Instance-based prediction [49] is conceptually different, as it does not explicitly construct a model, but rather picks the most likely route from the routes travelled in the past. While the main limitations are that instance-based cannot generalize to new routes, and may be computationally expensive to operate prototype search online, the main advantage is that it does not require extensive history for training, and can start predicting immediately after one route is completed by the user. Though it can than also only predict this one route as the second travel, over time this should converge to more sensible predictions.

Table 2: Overview of research work in route and destination prediction.

Paper	Year	Location	Goal	Data	Algorithm	Evaluation measure
[68]	2016	Bejing	Route pred.	Traces to network	PRoPTurn[b]	Accuracy and distance deviation
[56]	2016	Bejing	Route pred.	Traces to grid	NN, HMM	Cumulative likelihood
[47]	2015	Zhejiang[a]	Route patterns	Traces to polylines	SCPM[c]	Hausdorff distance
[70]	2015	Pisa	Dest. pred.	Traces to network	Myway	Accuracy and distance
[75]	2015	Bejing	Dest. pred.	Traces to grid	SubSyn [d]	Top-k (k=1,3) and distance error
						Baseline Bayesn traj. matching
[77]	2013		Dest. pred.	Traces to network		top-k (k=1,2,3)
			Route pred.			Over-trip link-overlap
[37]	2013	Seattle	Loc. pred.	Traces to network		Over-trip avg. rank-frac. of road
[58]	2013		Dest. pred.	Traces to network	HMM	Accuracy
			Route pred.		HMM, CPRM	Accuracy and Route distance
[76]	2013	Bejing	Dest. pred.	Traces to grid	SubSyn [d]	Top-k (over-trip)
			Route pred.			1-step accuracy
[28]	2013		Route pred.	Traces to network		(Within distance) accuracy
[55]	2012	Cambridge, MA	Dest. pred.	Traces to clusters	DPM[d]	Accuracy (over-trip + over-data)
[25]	2012		Dest. pred.	Traces to network	HDMN[e]	Accuracy
[14]	2011		Route pred.	Traces to grid	CPRM[b]	1-step acc. and Levenshtein dist.
[62]	2011	San Francisco	Loc. pred.	Traces to network	Nextplace[b]	Accuracy
[3]	2011	San Francisco	Loc. pred.	Traces to network	Nextplace[b]	Accuracy
[2]	2010	Sevilla[a]	Dest. pred.	Traces to network	HMM	Accuracy (over-trip)
[13]	2010		Dest. pred.	Traces to grid	Tree, HMM	Accuracy,
				Route pred.		1-step accuracy, Hausdorff dist.
[66]	2009		Dest. pred.	Traces to network	Several	Top-k (k=1,4)
[51]	2009	Milan	Loc. pred.	Traces to T-trees	WhereNext	Accuracy and Spatial Distance
[22]	2008	Seattle	Route pred.	Traces to polylines	Matching	Top-k (k=1,2,5,10)
[78]	2008	Bejing[a]	Dest. pred.	Traces to polylines	HMM, CPRM[b]	Accuracy
			Route pred.			Levenshtein, 1-step
[79]	2008	Pittsburg	Dest. pred.	Traces to network	Procab[f]	Accuracy and Likelihood
			Route pred.			Distance match., 90% accuracy
[69]	2007		Route pred.	Traces to clusters	Most likely	(not mentioned)
[43]	2007	Seattle	Dest. pred.	Traces to graph	HMM	Accuracy
[64]	2006	Detroit	Route pred.	Traces to network	HMM	1-step accuracy
[35]	2006	Seattle	Dest. pred.	Traces to grid	Bayesian	Accuracy (over-trip)
[38]	2006	Seattle	Dest. pred.	Traces to grid	Bayesian	(Within distance) accuracy
[41]	2006	Seattle	Route pred.	Traces to network	HMM	Accuracy
[32]	2003		Route pred.	Traces to network	Graph search	Accuracy
[5]	2003	Atlanta + Zurich	Dest. pred.		HMM	
[48]	2002		Dest. pred.		HMM, Histogram	Accuracy

[a] The authors do not explicity state the location, but the experiments seem based upon small internal data collection.
[b] Based upon clustering and a graph based prediction.
[c] Based upon PrefixSpan pattern mining algorithm [27].
[d] Based upon transition matrices.
[e] Based upon a combination of Bayesian networks & particle filters.
[f] Based upon Markov Decision Process and Inverse Reinforcement Learning

4 Evaluation criteria for mobility prediction

Setting up evaluation criteria for predictive systems of personal mobility may be challenging, since many different prediction scenarios and task settings are possible in different application scenarios.

When defining evaluation criteria, it is essential to select a criterion to capture the most relevant aspects of predictive performance for the application at hand, rather than selecting the most commonly used evaluation criteria for assessment. As a guideline, we identify the following six main prediction tasks.

1. Destination identification, identifying the regular destinations for a user (Section 4.1)
2. Location prediction (or estimation), predicting the location of the user a certain time ahead (Section 4.2)
3. Destination prediction, predicting the next destination of the user (Section 4.3)
4. Route prediction, predicting the route of the user to the next destination or location (Section 4.4)
5. Start-time prediction, although not mentioned specifically in the literature, would be to predict the next time the user is expected to leave his current location.
6. Travel-time prediction, to predict the time a travel will take, possibly dependent on route prediction.

We will focus on the first four tasks, which are concerned with location prediction. The last two tasks primarily focus on time prediction and are further from the main scope of our survey. Evaluation criteria for the latter two can be quite generic and therefore straightforward. One can opt, for instance, for the (Root) Mean Square Error (MSE) [74], the Mean Absolute Error or the Mean Absolute Percentage Error [23]. For a recent review on travel-time prediction see [54]. In general, the respective evaluation criteria need to capture the aspects that are relevant given the concrete purpose of the model. We will first identify the evaluation measures per task and then discuss the different trade-offs. Hat notation $\hat{y}$ is used for the prediction of predictor $P(\bullet)$, and thus the ground truth is simply y. The prediction $\hat{y}$ can be either a destination, location or a route depending on the task. Similarly, the input data X for the predictor can be one trace Tr or a set of traces

$$\hat{y} = P(X) = P\big((l_{t-x}, \dots, l_t)\big). \tag{4}$$

4.1 Destination identification

Destination identification is the task of identifying destinations from the traces of users. Such a setting can be used, for instance, in carpooling or recommending multi-modal transportation arrangements.

For each user u a list of destinations D^u, as a list with implicit order of importance. Then we denote d_k^u as the k-th entry of D^u with rank $\mathsf{rank}_D(d^u) = k$, and connected the

k-th most important destination, where importance is be defined either by total time spend or in number of visits. The latter is the most intuitive interpretation from a prediction perspective.

Identifying what is the destination from a given trace is not a trivial computational task. One may consider, for instance, whether bus stops count as destinations, or not, even though a person may be spending an extended time there. Alternatively, home may be a destination, even if very little time is spent there, which may be the case, for instance, when picking something or someone up and continuing to another destination. We believe that no technical solution without attaching semantics to the destinations would resolve this in a generic way applicable to all. Researchers are encouraged to explicitly indicate how their system handles such cases.

4.1.1 Subjective relevance

One can evaluate the destinations identified by asking the users of the application whether they consider them relevant. This leaves to the classic notions of false positives i.e. identified destinations which the user does not consider destinations and false negatives i.e. destinations the user considers relevant which were not identified in the data. A significant challenge with such a measure is to define in a universal way when a destination should be considered as important and when a place should be considered a destination in order to avoid such an evaluation being excessively user-dependent.

4.1.2 Predictive relevance

One can also evaluate the destinations identified (possibly as a function of b and r) by evaluating how it influences the error of the destination prediction (as discussed in 4.3). In other words, one can optimise the destination identification considering how it influences the error obtained in the prediction. Such optimisation of parameters should be done with careful separation of training data for parameter selection, another training data for model optimisation and hold out validation data for testing of the developed system.

We generally advice to handle destination definition and identification separately from predictive modelling, since the results here will influence the performance on the further dependent performance. Yet, as mentioned in the data section, it may be helpful to relate the performance of the system to the parameters r,b since these parameters determine what is identified as a set of destinations in a given dataset.

4.2 Location prediction

In location prediction, the task is to estimate the location l^u_{t+x} where the user u will be x minutes in the future, e.g. 30-minute-ahead prediction l^u_{t+30}. Note that this task does not depend on the identification of destinations, even though the terms location and destination sometimes are used interchangeably in the literature. Location prediction can be used, for example, for providing anticipated traffic jam information.

This task of location prediction has a stronger emphasis on time stamps of the mobility traces, since the goal is not only to know where the user is going, but also when he or she will be there. This is typically a shorter-horizon prediction/forecasting task than predicting a destination. This task may in fact be set up as a multi-target prediction task [71], where each location is predicted not stand alone, but as a part of a route sequence as in, for example, [59].

4.2.1 Distance to location in space

We can evaluate the discrepancy between the true location at time t and the predicted location with any distance measure. This can incorporate the square or exponential operator to punish larger errors more heavily

$$\mathsf{dist}(Y, \hat{Y}) = \frac{1}{N} \sum_{\hat{l} \in \hat{Y}, l \in Y} \mathsf{dist}(l_t, \hat{l}_t). \tag{5}$$

4.2.2 Time-independent distance

The main disadvantage of the distance measure is that it relies on travel time. Often it may be more informative to predict how close a user has been to a particular location, no matter when that happened. That is a valid scenario, among many, for instance if the objective of a predictions is to determine whether a user will see something (e.g. a fixed billboard). Then, at which time the user sees it comes secondary of the question whether it will be seen at all. In such a scenario for the location l_t at time t we take the closest location at time n where $t - w < n < t + w$ for a time-window size w

$$\mathsf{mindist}(Y, \hat{Y}) = \frac{1}{N} \sum_{\hat{l} \in \hat{Y}, Tr \in Y} \min_{l \in Tr} \left| \mathsf{dist}(l_n, \hat{l}_t) \right|. \tag{6}$$

Naturally, the time and space distance can also be combined in any fashion into a single criteria.

4.3 Destination prediction

A trace may include multiple destinations. The task of predicting a destination is to predict the next location given the location now, and possibly some contextual information. It is important to specify whether the predicted destination is a point in space or a point in time-space.

4.3.1 (Binary) Accuracy

For a trace of length N, predict the destination $\hat{d}$: How often is the predicted destination indeed correct? This gives an accuracy of the prediction. As any classification problem, other criteria like sensitivity and fall-out can also be used.

$$\mathsf{accuracy}(Y, \hat{Y}) = \frac{1}{N} \sum_{\hat{d} \in \hat{Y}, d \in Y} 1_{d = \hat{d}}. \tag{7}$$

We note that this case bears similarity to the task of multi-label classification. In this scenario (see [71] for a review), binary accuracy is known as Hamming loss. A variety of evaluation metrics are used for multi-label evaluation (including ranking metrics, which we discuss next) though Hamming loss remains one of the most popular, especially if outputs (destinations, in this case) are presumed to be predicted independently.

4.3.2 Weighted accuracy

It could also be the case that the prediction mechanism outputs a probability for each prediction. Then the error could be weighted on this probability, giving for example log loss.

$$\mathsf{logloss}(Y, \hat{Y}) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)). \tag{8}$$

Often in implementations of log loss the predictions are bounded away from the extremes by a small value.

4.3.3 Rank or top-k

Often the predictive systems are designed to output a ranked list of destinations $\hat{D}$ (with the rank based on e.g. probability values). One way to assess the quality of the prediction is to check how often the correct destination is among the top-k predictions

$$\mathsf{top}\text{-}k(Y, \hat{Y}) = \frac{1}{N}\sum_{\hat{D}\in\hat{Y}, d\in Y} 1_{\mathrm{rank}_{\hat{D}}(d)\leq k}. \tag{9}$$

Note that accuracy above can be considered a special case of top-k with $k = 1$. In scenario from 2.1, the user was presented four options in the route planning application; top-4 then is an intuitive evaluation measure to use.

Another metric is the average rank of the prediction. This has as disadvantage that it depends on the scale of the prediction (the error is expected to increase for longer $\hat{D}$).

$$\mathsf{avgrank}(Y, \hat{Y}) = \frac{1}{N}\sum_{\hat{D}\in\hat{Y}, d\in Y} \mathrm{rank}_{\hat{D}}(d). \tag{10}$$

4.3.4 Accuracy in space

When destination prediction in space (coordinates) is output, one may assess the accuracy by asking how far the predicted destination is from the true destination. This can be measured by a distance measure, such as the MSE, in coordinate space. The major challenge is sensitivity of this approach to large deviations or outliers. This may happen due to, for instance, occasional trips to other cities for example. See (5). Therefore, one needs to consider how to ensure robustness of such measures. One way to do that is to put a cap on large deviations, essentially removing the outliers of the individual errors before taking the mean.

4.3.5 Baseline accuracy

No matter which accuracy measure is used, it is critical to consider what would be the naive baseline performance, as otherwise measured accuracy may be non-informative or even misleading. In many cases baseline accuracy can be derived from so called Origin-Destination (OD) transition matrix, which describes in probabilities how often the user transfers from one destination to another. This directly relates to the first order Hidden Markov Models. To be clear what the added value of a system recording GPS traces is, one is strongly recommended to asses the performance relative to such a baseline, since measured absolute accuracy depends on how *easy* the patterns in the data are to discover.

4.4 Route prediction

The task of predicting a route is to, given the recent trace, predict a sequence of locations that will be the future trace. Route prediction is typically done using discrete waypoint sequences rather than raw GPS traces [16, 49].

4.4.1 Next-up or one-step accuracy

Next-up or one-step accuracy is often used in waypoint prediction. It describes, how often is the next waypoint predicted correctly. Since each waypoint – generally a road intersection – is only connected to a few other waypoints, this leads to very high accuracy numbers. However, it is more one-step-ahead destination prediction than actual route prediction. To consider: if the accuracy is 90 percent, then the chance of predicting a route of 30 waypoints correct is merely $0.9^{30} = 0.04$. The formula is analog to (11):

$$\mathsf{1\text{-}step}(Y, \hat{Y}) = \frac{1}{N} \sum_{\hat{w} \in \hat{R} \in \hat{Y}, w \in R \in Y} 1_{w=\hat{w}}. \tag{11}$$

4.4.2 Ranking methods

Just as with destination prediction, route prediction is sometimes ranked. Again the same method applies as with (9) and (10). For convenience we denote a collection of routes as $S = \{R_1, ..., R_N\}$.

$$\mathsf{avgrank}(Y, \hat{Y}) = \frac{1}{N} \sum_{\hat{S} \in \hat{Y}, r \in Y} \mathsf{rank}_{\hat{S}}(r). \tag{12}$$

4.4.3 Matching methods

Another method is to match either traces or waypoints. There are many common methods that could be used here (common subsequence, string matching, DTW). We will focus on two common methods, the Hausdorff distance for grid structures (or curves) and the Levenshtein distance for waypoint sequences.

The asymmetric Hausdorff distance $\mathsf{dist}_H(x, y)$ is often used for either a curve or grid representation of the data. Due to its asymmetry, this returns precision accuracy for the distance $\mathsf{dist}_H(\hat{R}, R)$ of the predicted route to the real route and prediction completeness

$\mathsf{dist}_H(R, \hat{R})$ for the distance of the real route to the predicted route. The asymmetry can be solved by taking the max of the two values—as in the formula below. A variant of this metric is used in [47].

$$\mathsf{dist}_H(Y, \hat{Y}) = \max\{\sup_{y \in Y} \inf_{\hat{y} \in \hat{Y}} \mathsf{dist}(y, \hat{y}), \sup_{\hat{y} \in \hat{Y}} \inf_{y \in Y} \mathsf{dist}(y, \hat{y})\}. \tag{13}$$

Levenshtein distance is also known as the edit-distance from string matching. It indicates how many changes need to be made between two sequences in order to become fully equal. It works well with routes that are represented as a sequence of waypoints.

$$\mathsf{dist}_L(Y, \hat{Y}) = \mathsf{lev}_{Y,\hat{Y}}(\|Y\|, \|\hat{Y}\|) \tag{14}$$

$$\mathsf{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} \mathsf{lev}_{a,b}(i-1,j) + 1 \\ \mathsf{lev}_{a,b}(i,j-1) + 1 \\ \mathsf{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases}$$

4.4.4 Baseline accuracy

Baseline prediction of routes could be the most common route given the destination (and time). Methods of accuracy could be compared to this baseline. For example when using top-k rank accuracy, it might be interesting to mention the naive accuracy of predicting the k most common routes.

4.4.5 Fraction-of-trip

Another dimension that can be taken into account is the destination prediction as a fraction of the trip completed—this is done in e.g. [22, 2]. The idea is that when one gets closer to his or her destination, it will be increasingly easy to predict. This measure can be incorporated for each of the above criteria, and is indeed sensible to compute. However, since it is computed as a fraction of the entire trip, it can only be computed in retrospect and is very sensitive to e.g. extremely long (very difficult for a long time) or extremely short (very easy all the time) trips. Although graphs of fraction-of-trip errors are illustrating, they are hard to compare between different papers. It would be valuable if the authors mentioned the area under the ROC curve (AUROC) [9], because it translates the plot to a number that enables comparison with other methods. Another method to allow for cross-study comparison is to specify top-k accuracy for prediction on e.g. 30% and 70% of the trip—see e.g. [75].

4.4.6 On-the-fly evaluation

As mentioned, the fraction-of-trip error cannot be provided before the length of the trip is known—i.e. before the trip is completed. For on-the-fly evaluation, one will have to rely on other evaluation criteria. For example, matching or one-step accuracy, can be applied to partial routes.

Table 3: Descriptive table for the GPS trace data of the TrafficSense project used in the experiments of this paper.

Users	15
Users affiliation	Research group
Day per users	194 days between 10-11-2015 & 23-05-2016
Sensor	Phone sensors
Method of collection	Custom client for Android
Location	Helsinki region
Data	GPS traces (lon, lat, acc)
Data interval	Irregular intervals; depending on movement and activity
Contextual data	Time. Activity recognition.

4.4.7 Multi-user situations

Evaluating a system that has multiple users requires aggregation and averaging, the choice of which is not trivial. Individual errors can be aggregated over all the users, or over all the trips. In the first case, users with more irregular patterns will cause the error to move up; as in the second case, users with more data will both be easier to predict and have a larger share of the total amount of trips. One interesting statistic for analysis of the system performance could be to show the per user error as a function of data, which would indicate necessary and sufficient training time for an accurate system, as performed, for example, in [55].

5 Experimental analysis

The goals of this experimental analysis are to complement the survey of measures by empirical evidence, provide illustration of selected concepts, highlight practical challenges, and provide recommendations. We do not perform separate experiments on location prediction, as the evaluation measures are relatively straightforward compared to destination and route prediction. The experimental study uses original and previously unpublished mobility data collected from 15 users over the period of half a year in the Helsinki metropolitan area, as summarised in Table 3. The code was written in Python with the plots provided by matplotlib [31]. All the experiments are performed separately for the data of each user, mimicking privacy situations for possible implementations.

5.1 Destination identification

Our first experiment focuses on destination identification, as it underlies the later prediction problems. This method should find the places in which the users spends b seconds in a (circular) range of r meters. In line with the general method, we use a sliding maximal window of at most r distance end-to-end, where a stop is marked on the range if the time end-to-end reaches or exceeds b. Overlapping stop ranges are combined into a single stop. It results in a list of destinations D, including how often it was visited, the respective trace

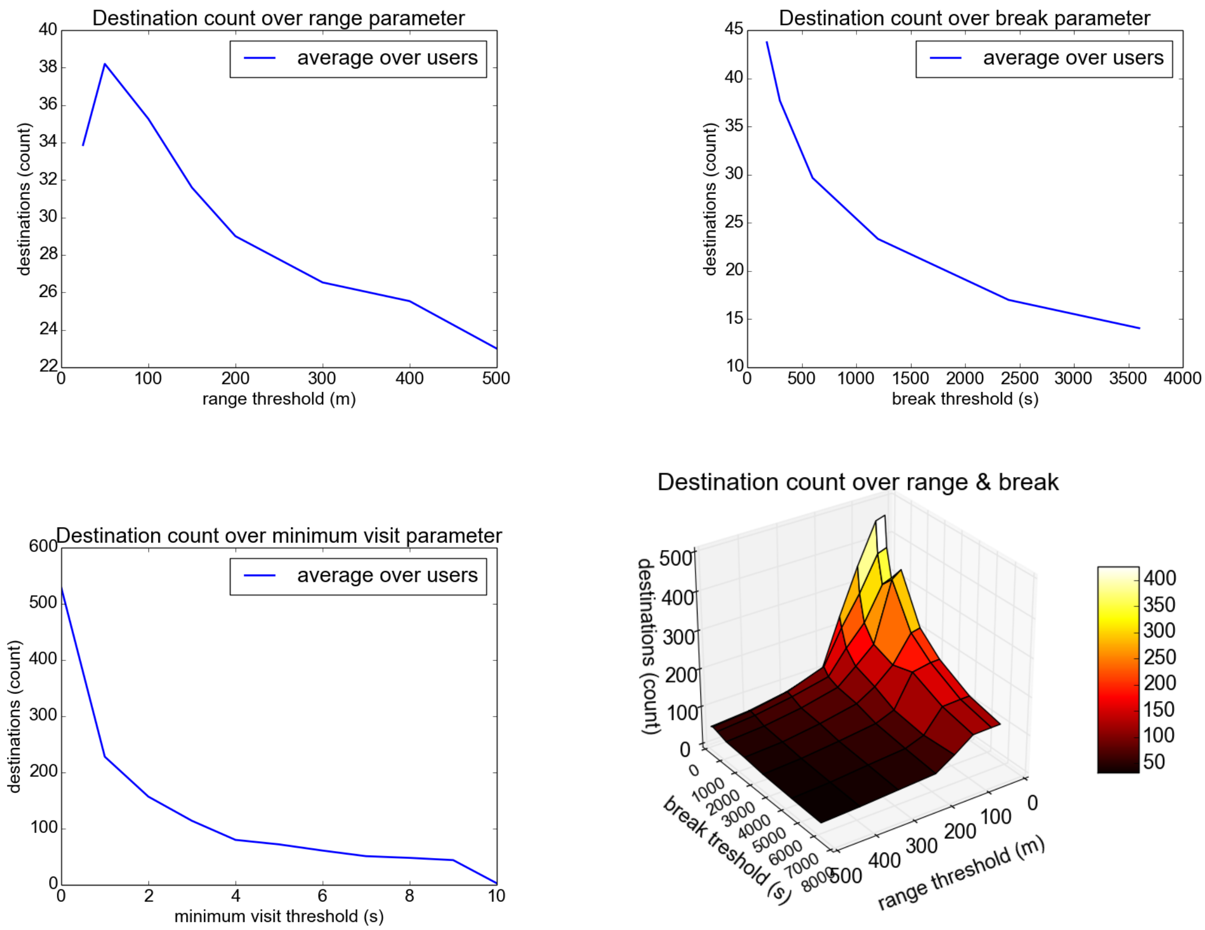

Figure 3: Destination count as a function of thresholds.

of this visit, and how much time was spent there. This extra information enables us to rank the destinations, first by amount of visits and then by time.

The plots in Figure 3 illustrate the difference in the number of destinations given the parameters for minimum time spend b, the minimum range r and the minimum amount of visits required. One can observe that the amount of destinations identified falls for each parameter as destinations that are close together get combined into one single destination. The only irregularity is found for a small range. Below a certain range less destinations are identified since the user has too few points to still make it count as a destination.

In some sense we are enforcing a minimal speed by setting these parameters together - below this speed, the user is considered to be at a place rather than moving past it. The 3D plot illustrates the connection of the range and break parameter. Clearly, the number of destinations falls much more quickly with respect to range r than it does for time b. As discussed, the parameter setting depends on one's definition for destination, which in turn depends on the requirements of the system. It seems most stable to take a parameter combination somewhere in the plane of the 3D plot, which is indeed done by $r = 200, b = 300$ [2] or $r = 320, b = 600$ [5].

An important criteria for system operation is how much data is required in order to learn

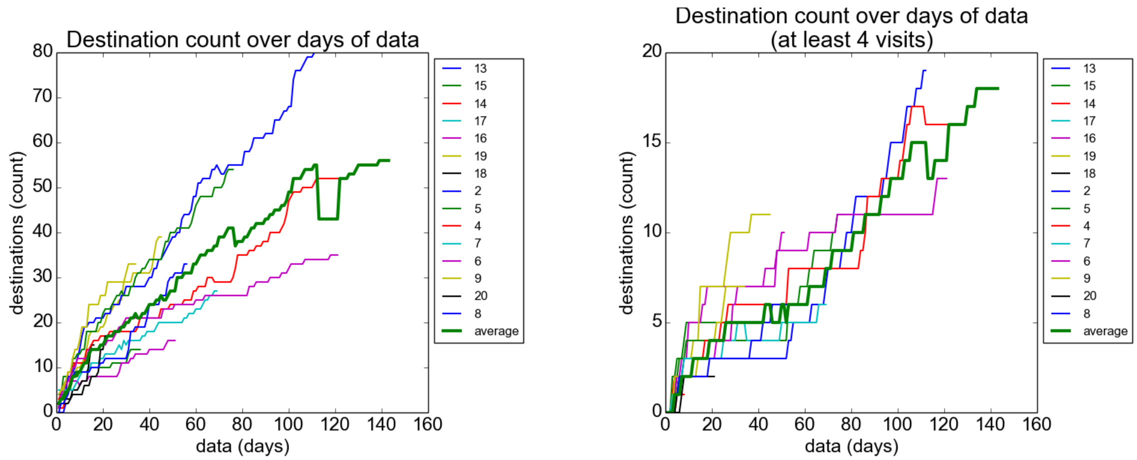

Figure 4: Destination count over days.

to predict the behaviour of its users reasonably well. We analyse this by plotting the number of destinations as a function of the amount of days in Figure 4. The left panel in Figure 4 has no requirement on how often a user needs to return, whereas the right panel in Figure 4 enforces such a pattern. Human patterns are likely to be regular and predictable [26]. Yet in both plots the number of destinations steadily continues increasing. Thus, it looks like the amount of regular destinations is much larger than we might initially suspect and was not saturated in the current data collection period, or travel patterns keep changing over time such that there is no clear saturation. Our analysis identifies destinations in a purely data-driven way and autonomously from any semantics of the destination. Even with the most inclusive parameter settings we get less than 40 destinations on average per user over two-three months. Intuitively, the destination counts look plausible and prudent (keeping in mind that this should cover work, home, various shops, meeting, public event places and such). Most likely, destination counts do not explicitly saturate in our plots, because the data collection period is not long enough to make it notable. This has an important implication to rigorous performance evaluation. Performance measures need to have a forgetting mechanisms in order to be able to accommodate and distinguish between initial data collection phase and mature system operation.

As mentioned, subjective evaluation has several drawbacks due to the semantics discussion about importance and destinations. One could also see r and b as hyper parameters of the prediction model. In Figure 5 we show the accuracy of a simple tree predictor over the parameter grid. will discuss more about the actual prediction. The results shown are on the last 40% of each users days, thus an independent test set. The plot accurately captures the main draw back: for high range parameters some destinations will be merged—as they are close together—and thus prediction accuracy will increase.

5.2 Destination prediction

To illustrate the measures of destination prediction in more detail, we implement a naive baseline and a more sophisticated predictor. As a baseline, we take the intuitive method of predicting the most likely destination given someone's current position. This is done by the

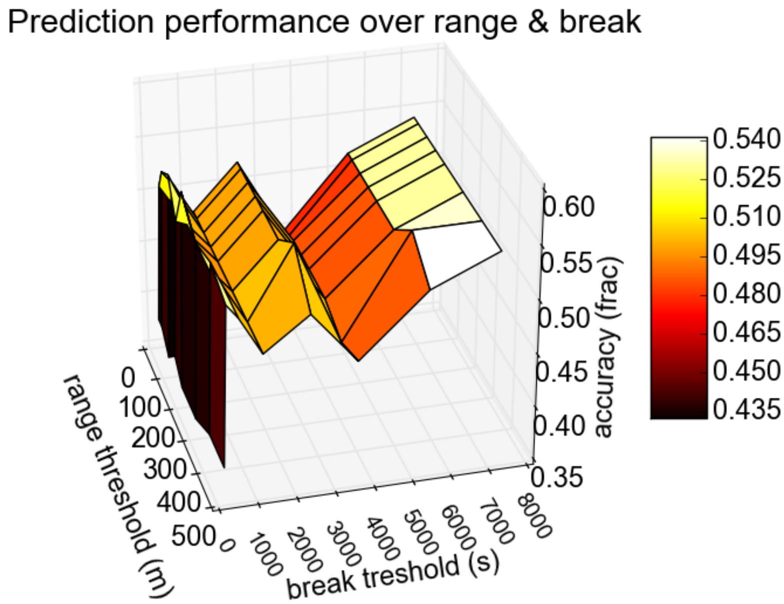

Figure 5: Prediction accuracy with different parameter selection.

most simple incarnation of a Markov Chain which simply counts the transitions from node to node. For the classifier, we use a classification tree as it is merely an illustration of the trends in evaluation measures. We use the implementation of sci-learn [57].

We plot the accuracy of this baseline for each user—see Figure 6. Observe that the accuracy is higher for a subset gof three users; probably they have a more regular travel pattern. We use this baseline to compare the performance of a (only slightly more sophisticated) classification tree that receives contextual information about day of the week and time of the day. Here the performance increase slows down significantly, after around 10 days for the simple predictor and 20 days for the classification tree. Still the performance does not fully saturate.

We also illustrate how k-accuracy increases for higher values of k. The increase, as expected, slowly saturates because a part of the error results from transfers from or to destinations not observed within the training data. To show their performance, authors could provide a similar plot [22] or use the recently common $k = 1, 3$ [75, 77].

We have discussed that the error in systems can be averaged either per user or per trip. In Figure 7 we illustrate the difference between these two different approaches. One can see that the trip average (blue) shows a higher number, depending of course on the amount of users present. This should be taken into account when judging the performance of a system: providing trip averages will likely show a higher performance, but not congruent with the goal of most applications which will want to offer a good performance to each of their users.

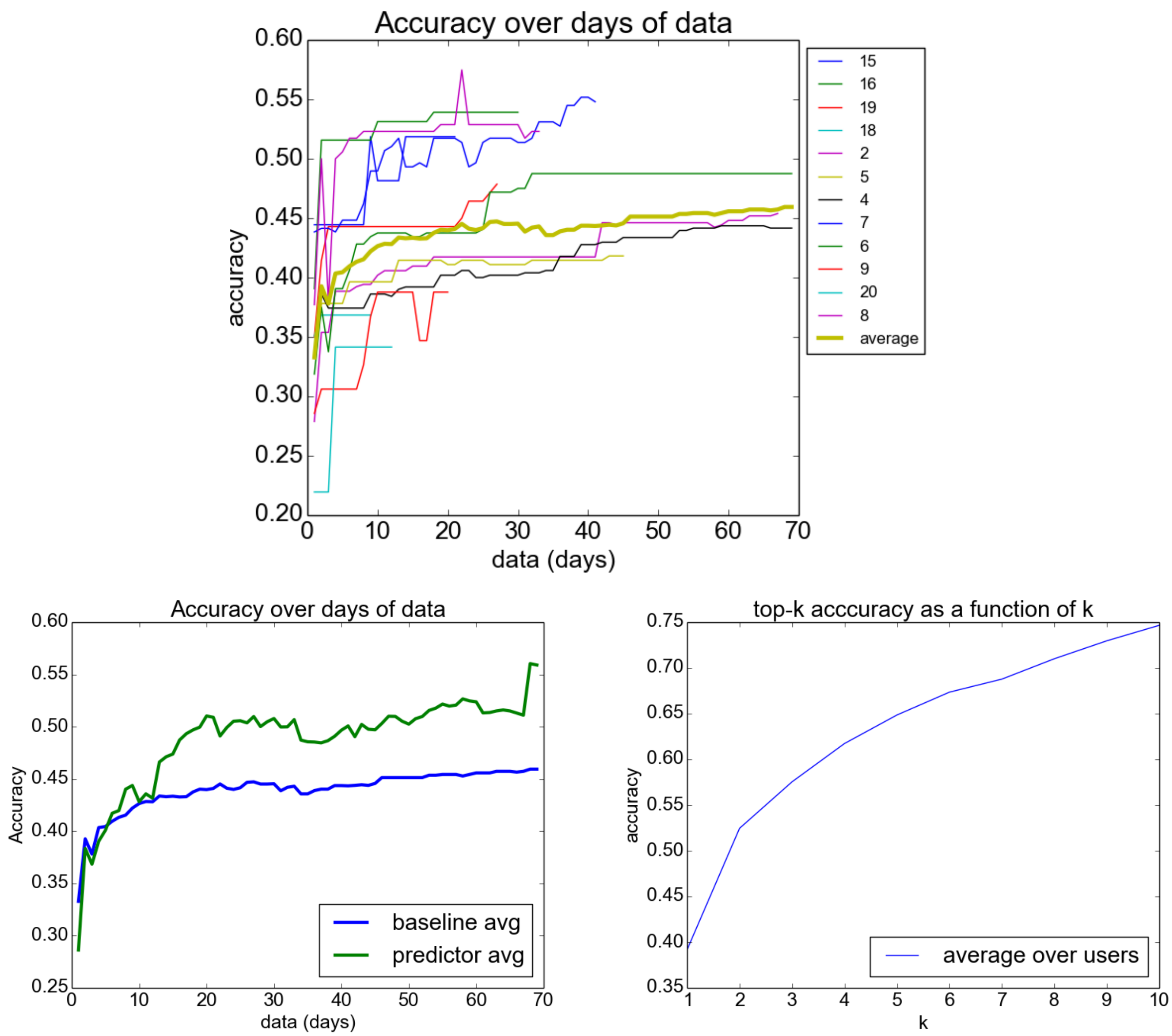

Figure 6: Accuracy of destination prediction.

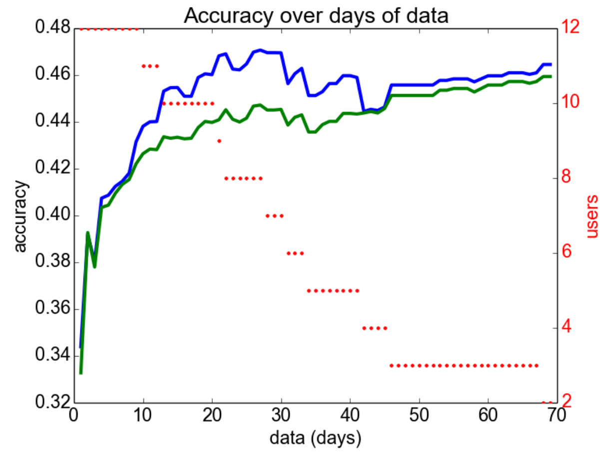

Figure 7: Accuracy of user-average (blue) vs. trip-average (green). Red dots indicate the number of users, denoted on the right axis.

5.3 Route prediction

As final illustration we show the effect of different error measurements on route prediction. For that purpose we generate instance-based route predictions in the following controlled manner, which allows us for analysis purpose to control the percentage of routes that are predicted incorrectly, and investigate, how well the evaluation measures capture this.

The routes are deducted from the traces, where each route is simply the part between two visits of a destination, filtering out trips that last for more than 3 hours as someone might have had his phone switched off. The waypoint representation traces are snapped to the closest road intersection provided by OpenStreetMap, in which waypoint is a unique integer.

First we take all the routes and permute a fixed percentage of the routes over time (e.g. swap Monday and Wednesday). We take the mean over different random seeds in order to balance the impact of the seed. In Figure 8 we show how different measures of accuracy differ over the percentage of permuted routes. As expected, one-step accuracy is much higher than normal accuracy that closely follows the diagonal. Note that in this system entire routes are permuted; a real-time one-step prediction will likely have even higher accuracy as from each waypoint only a few transitions are possible.

6 Conclusions

Our analysis and experiments have demonstrated that the observed accuracy is very dependent on the users and averaging over the users. This has implications for designing and interpreting evaluation systems – we recommend the base for averaging to be closely related to the expected user base. We have shown top-k accuracy analysis, and in research recommend to report at least $k = 1, 3$. In practice, of course, the choice depends on the implementation, where an application might be able to show several recommendations to the user; in that case k should match the number of recommendations. Furthermore, our exper-

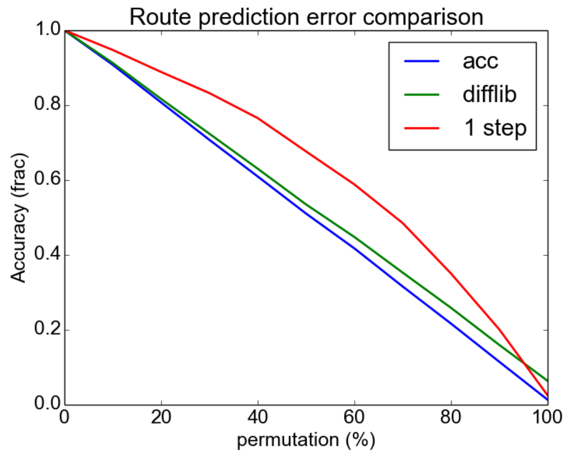

Figure 8: Accuracy of route prediction.

iments have shown that varying interpretations may be obtained depending on whether the accuracy is averaged over users or over trips. We have argued that trip-average will bias the results towards more predictable and more active users, therefore, one should consider the underlying purpose of the predictive system when making this choice. Finally, we have emphasised and experimentally illustrated that whichever measures are selected, in both route and destination prediction it is critical to report and compare to a naive baseline. Especially because travel patterns tend to be highly imbalanced, in which case accuracy would tend to show promising figures, which are likely to be due to easiness or difficulty of the prediction task for a specific user rather than merits of a chosen computational approach.

Our main recommendation for researchers and practitioners developing approaches for personalised mobility analysis is to consider user-specific accuracy measures in addition to averaged aggregates, as well as to take into consideration that for many users accuracy does not saturate fast and the performance keeps evolving over time, therefore, we recommend using time-sensitive measures.

Acknowledgements

This work was supported by the TrafficSense research project, funded by the Aalto University AEF research programme `http://energyefficiency.aalto.fi/en/`. The authors would like to thank the participants of the TrafficSense research project for their feedback during preparation of this manuscript. Special acknowledgement to Mikko Rinne, Seppo Tormä, and Kimmo Karhu for developing and maintaining the data collection client software and the connected road network snapping throughout the project.

References

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[2] J. A. Alvarez-Garcia, J. A. Ortega, L. Gonzalez-Abril, and F. Velasco. Trip destination prediction based on past gps log using a hidden markov model. *Expert Systems with Applications*, 37(12):8166–8171, 2010.

[3] Th. Anagnostopoulos, Ch. Anagnostopoulos, and S. Hadjiefthymiades. Mobility prediction based on machine learning. In *Proc. of the 12th IEEE International Conference on Mobile Data Management*, volume 2, pages 27–30, 2011.

[4] N. Andrienko and G. Andrienko. A visual analytics framework for spatio-temporal analysis and modelling. *Data Mining and Knowledge Discovery*, 27(1):55–83, 2013.

[5] D. Ashbrook and Th. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[6] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[7] Y. Bengio. Markovian models for sequential data. *Neural Computing Surveys*, 2:129–162, 1999.

[8] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day series in time series analysis and digital processing. Holden-Day, San Francisco, 1976.

[9] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[10] L. Cao and J. Krumm. From gps traces to a routable road map. In *Proc. of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 3–12, 2009.

[11] Ch. Chen, D. Zhang, Zh. H. Zhou, N. Li, T. Atmaca, and Sh. Li. B-planner: Night bus route planning using large-scale taxi gps traces. In *Proc. of IEEE International Conference on Pervasive Computing and Communications*, PerCom, pages 225–233, 2013.

[12] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[13] L. Chen, M. Lv, and G. Chen. A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 6(6):657–676, 2010.

[14] L. Chen, M. Lv, Q. Ye, G. Chen, and J. Woodward. A personal route prediction system based on trajectory data mining. *Information Sciences*, 181(7):1264–1284, 2011.

[15] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proc. of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090, 2011.

[16] J. Dai, B. Yang, Ch. Guo, and Zh. Ding. Personalized route recommendation using big trajectory data. In *Proc. of the 31st IEEE International Conference on Data Engineering*, ICDE, pages 543–554, 2015.

[17] Y. A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3, 2013.

[18] Y. Deguchi, K. Kuroda, M. Shouji, and T. Kawabe. HEV charge/discharge control system based on navigation information. Technical report, SAE Technical Paper, 2004.

[19] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction*, 16(2):97–166, 2001.

[20] K. L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer Publishing Company, 2013.

[21] E. Ericsson, H. Larsson, and K. Brundell-Freij. Optimizing route choice for lowest fuel consumption–potential effects of a new driver support tool. *Transportation Research Part C: Emerging Technologies*, 14(6):369–383, 2006.

[22] J. Froehlich and J. Krumm. Route prediction from trip observations. Technical report, SAE Technical Paper, 2008.

[23] X. Gang, W. Kang, F. Wang, F. Zhu, Y. Lv, X. Dong, J. Riekki, and S. Pirttikangas. Continuous travel time prediction for transit signal priority based on a deep network. In *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, pages 523–528, 2015.

[24] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Röning, and J. Riekki. Personalised assistance for fuel-efficient driving. *Transportation Research Part C: Emerging Technologies*, 58:681–705, 2015.

[25] V. Gogate, R. Dechter, B. Bidyuk, C. Rindt, and J. Marca. Modeling transportation routines using hybrid dynamic mixed networks. *arXiv*, 1207.1384, 2012.

[26] M. C. Gonzalez, C. A. Hidalgo, and A. L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453:779–782, 2008.

[27] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. of the 17th international conference on data engineering*, pages 215–224, 2001.

[28] P. Helmholz, E. Ziesmann, and S. Robra-Bissantz. Context-awareness in the car: prediction, evaluation and usage of route trajectories. In *Proc. of International Conference on Design Science Research in Information Systems*, pages 412–419, 2013.

[29] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[30] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Achieving guaranteed anonymity in gps traces via uncertainty-aware path cloaking. *IEEE Transactions on Mobile Computing*, 9(8):1089–1107, 2010.

[31] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[32] H. A Karimi and X. Liu. A predictive location model for location-based services. In *Proc. of the 11th ACM international symposium on Advances in geographic information systems*, pages 126–133, 2003.

[33] L. M. Kieu, A. Bhaskar, and E. Chung. Transit passenger segmentation using travel regularity mined from smart card transactions data. *Transportation Research Board 93rd Annual Meeting*, 2014.

[34] J. M. Kim, H. J. Beak, and Y. T. Park. Incremental clustering method for recognizing user destinations and routes using smartphone global positioning system sensor. *Sensors and Materials*, 27(8):599–604, 2015.

[35] J. Krumm. Real time destination prediction based on efficient routes. Technical report, SAE Technical Paper, 2006.

[36] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[37] J. Krumm, R. Gruen, and D. Delling. From destination prediction to route prediction. *Journal of Location Based Services*, 7(2):98–120, 2013.

[38] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *Proc. of the International Conference on Ubiquitous Computing*, pages 243–260, 2006.

[39] A. R. Kuppam and R. M. Pendyala. A structural equations analysis of commuters' activity and travel patterns. *Transportation*, 28(1):33–54, 2001.

[40] K. Laasonen. Route prediction from cellular data. In *Proc. of Workshop on Context-Awareness for Proactive Systems*, volume 1617, 2005.

[41] J. Letchner, J. Krumm, and E. Horvitz. Trip router with individualized preferences (TRIP): Incorporating personalization into route planning. In *Proc. of of the 18th conf. on Innovative applications of artificial intelligence*, IAAI, pages 1795 – 1800, 2006.

[42] D. Levinson and A. Kumar. Activity, travel, and the allocation of time. *Journal of the American Planning Association*, 61(4):458–470, 1995.

[43] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5):311–331, 2007.

[44] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11, 2003.

[45] M. Lin, W. J. Hsu, and Z. Q. Lee. Predictability of individuals' mobility with high-resolution positioning data. In *Proc. of the ACM Conference on Ubiquitous Computing*, pages 381–390, 2012.

[46] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu. Mining large-scale, sparse gps traces for map inference: comparison of approaches. In *Proc. of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 669–677, 2012.

[47] M. Lv, Y. Li, Zh. Yuan, and Q. Wang. Route pattern mining from personal trajectory data. *Journal of Information Science and Engineering*, 31(1):147–164, 2015.

[48] N. Marmasse and Ch. Schmandt. A user-centered location model. *Personal and ubiquitous computing*, 6(5-6):318–321, 2002.

[49] O. Mazhelis, I. Zliobaite, and M. Pechenizkiy. Context-aware personal route recognition. In *Proc. of International Conference on Discovery Science*, pages 221–235, 2011.

[50] J. Mendes-Moreira, L. Moreira-Matias, J. Gama, and J. F. de Sousa. Validating the coverage of bus schedules: a machine learning approach. *Information Sciences*, 293:299–313, 2015.

[51] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 637–646, 2009.

[52] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, 2013.

[53] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. Predicting taxi-passenger demand using streaming data. *IEEE Trans. Intelligent Transportation Systems*, 14(3):1393–1402, 2013.

[54] U. Mori, A. Mendiburu, M. Álvarez, and J. A. Lozano. A review of travel time estimation and forecasting for advanced traveller information systems. *Transportmetrica A: Transport Science*, 11(2):119–157, 2015.

[55] A. Nadembega, T. Taleb, and A. Hafid. A destination prediction model based on historical data, contextual knowledge and spatial conceptual maps. In *Proc. of IEEE International Conference on Communications*, ICC, pages 1416–1420, 2012.

[56] Ph. Pecher, M. Hunter, and R. Fujimoto. Data-driven vehicle trajectory prediction. In *Proc. of the annual ACM Conference on Principles of Advanced Discrete Simulation*, SIGSIM, pages 13–22, 2016.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[58] D. Qiu, P. Papotti, and L. Blanco. Future locations prediction with uncertain data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–432, 2013.

[59] J. Read, L. Martino, and J. Hollmén. Multi-label methods for prediction with sequential data. *Pattern Recognition*, 63:45–55, 2017.

[60] J. Read, I. Žliobaitė, and J. Hollmén. Labeling sensing data for mobility modeling. *Information Systems*, 57:207–222, 2016.

[61] M. Roth, F. Gustafsson, and U. Orguner. On-road trajectory generation from gps data: a particle filtering/smoothing application. In *Proc. of the 15th International Conference on Information Fusion*, FUSION, pages 779–786, 2012.

[62] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *International Conference on Pervasive Computing*, pages 152–169, 2011.

[63] D. E. Seidl, P. Jankowski, and M. H. Tsou. Privacy and spatial pattern preservation in masked gps trajectory data. *International Journal of Geographical Information Science*, 30(4):785–800, 2016.

[64] R. Simmons, B. Browning, Y. Zhang, and Varsha Sadekar. Learning to predict driver route and destination intent. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 127–132. IEEE, 2006.

[65] Ch. Song, Z. Qu, N. Blumm, and A. L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[66] K. Tanaka, Y. Kishino, T. Terada, and Sh. Nishio. A destination prediction method using driving contexts and trajectory for car navigation systems. In *Proc. of the ACM symposium on Applied Computing*, pages 190–195, 2009.

[67] E. D. Tate and S. P. Boyd. Finding ultimate limits of performance for hybrid electric vehicles. Technical report, SAE Technical Paper, 2000.

[68] F. Terroso-Saenz, M. Valdes-Vela, and A. F. Skarmeta-Gomez. Online route prediction based on clustering of meaningful velocity-change areas. *Data Mining and Knowledge Discovery*, pages 1–40, 2016.

[69] K. Torkkola, K. Zhang, H. Li, H. Zhang, Ch. Schreiner, and M. Gardner. Traffic advisories based on route prediction. In *Proceedings of Workshop on Mobile Interaction with the Real World*, pages 33–36, 2007.

[70] R. Trasarti, R. Guidotti, A. Monreale, and F. Giannotti. Myway: Location prediction via mobility profiling. *Information Systems*, 2015.

[71] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*. 2nd edition, Springer, 2010.

[72] I. Žliobaitė, A. Bifet, J. Read, B. Pfahringer, and G. Holmes. Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning*, 98(3):455–482, 2014.

[73] I. Žliobaitė and J. Hollmén. Mobile sensing data for urban mobility analysis: A case study in preprocessing. In *Proc. of the Mining Urban Data Workshop at EDBT/ICDT*, pages 309–314, 2014.

[74] Ch. H. Wu, J. M. Ho, and D. T. Lee. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281, 2004.

[75] A. Y. Xue, J. Qi, X. Xie, R. Zhang, J. Huang, and Y. Li. Solving the data sparsity problem in destination prediction. *The VLDB Journal*, 24(2):219–243, 2015.

[76] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang, and Zh. Xu. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *Proc. of IEEE 29th International Conference on Data Engineering*, ICDE, pages 254–265, 2013.

[77] T. Yanagihara, R. Namiki, K. Nawa, D. Weir, and K. Oguchi. Combining prediction methods with cyber information for proactive route recommendation. In *Proc. of the IEEE 3rd Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, CYBER, pages 87–91, 2013.

[78] Q. Ye, L. Chen, and G. Chen. Predict personal continuous route. In *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems*, pages 587–592, 2008.

[79] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 322–331, 2008.

[80] W. Zucchini and I. L. MacDonald. *Hidden Markov models for time series: an introduction using R*, volume 150. CRC press, 2009.