Estimation of a recursive link-based logit model and link flows in a sensor equipped network

van Oijen, Tim P.; Daamen, Winnie; Hoogendoorn, Serge P.

DOI
10.1016/j.trb.2020.08.003

Publication date
2020

Document Version
Final published version

Published in
Transportation Research Part B: Methodological

Transportation Research Part B 140 (2020) 262–281

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Estimation of a recursive link-based logit model and link flows in a sensor equipped network

Tim P. van Oijen*, Winnie Daamen, Serge P. Hoogendoorn

Department of Civil Engineering and Geosciences, Delft University of Technology, Transport & Planning, PO Box 5048, Delft, 2600 GA, the Netherlands

ARTICLE INFO

Article history:
Received 23 September 2019
Revised 19 April 2020
Accepted 26 August 2020

Keywords:
Discrete choice modeling
Recursive link-based logit model
Wi-Fi-sensors
Crowd monitoring
Route choice
Link flow estimation

ABSTRACT

This paper describes a method to estimate the parameters of a Recursive link-based Logit model (RL) using measurements of a set of spatially fixed proximity sensors, with limited hit rates, which can uniquely identify people, such as Wi-Fi-, RFID- or Bluetooth-sensors. The observed 'route' of an individual, where we focus on pedestrians in an urban or event context, is modelled as the sequence of sensors that have identified the individual during his or her trip. Obviously, these 'routes' contain large gaps, which makes traditional estimation techniques not applicable. Although we do not exactly know what happens within these gaps, we do have some specific insight about the individuals behavior between two identifications; we know with a certain probability which is related to the hit rate of the sensors, that the individual did not cross another sensor location between the two identifications. This paper therefore describes a method to estimate the parameters of an RL model that specifically exploits this knowledge. The framework also allows us to formulate a probabilistic link utilization estimation method, which can be used to estimate link flows in a network based on the sensor observations. The effectiveness of the methodology is demonstrated in simulation using an artificial network, after which the methodology is tested on a real data set, collected at a Dutch music event.

1. Introduction

Discrete choice models have been used for decades to describe all kinds of human mobility, including activity choice, destination choice, mode choice and route choice. For the latter, which has the main focus in this paper, many models have been successfully used, like Probit, Multinomial Logit (MNL), C-Logit (Cascetta et al., 1996), Path-Size Logit (Ben-Akiva and Bierlaire, 1999) and Recursive link-based Logit (RL) (Fosgerau et al., 2013; Mai et al., 2018). In the RL model, which has a graph-based representation, route-choice is modeled as sequentially choosing a next link. We will later see that this structure is also very suitable for our sensor-based estimation method.

Regardless of the exact model type, data is required to estimate the parameters of a discrete choice model. A large amount of data sources can be adopted, like GPS-traces, Bluetooth-traces, on-line or off-line surveys, Wi-Fi-traces, RFID, and mobile network phone data. Although each type of data comes with their specific characteristics when it comes to accuracy,

* Corresponding author.
E-mail addresses: T.P.vanOijen@tudelft.nl (T.P. van Oijen), W.Daamen@tudelft.nl (W. Daamen), S.P.Hoogendoorn@TUDelft.nl (S.P. Hoogendoorn).

https://doi.org/10.1016/j.trb.2020.08.003

availability and processing, we can make a very clear distinction between two types of data sources: location data sources and proximity data sources. Location data, like GPS-traces and self-reported routes, consists of measured or reported locations of people. Proximity data, like Wi-Fi-, Bluetooth- or RFID-traces, consists of sequences of fixed locations (sensors) where a certain person was observed. At first glance, these two data sources look very similar, but if we think a bit longer, we realize that there is a fundamental difference between the two types of data. From location data, we do not get any insight into what happened between two consecutive measurements. With proximity data, however, we know that it is impossible (or very unlikely) that a person passed sensor locations elsewhere in the network between two consecutive sensor observations. In other words, with proximity data, contrary to location data, we do have insight into what (most likely) did not happen between two consecutive measurements. Apparently, estimation of a route choice model using proximity data asks for its own approach, especially since the gaps between two sensor observations can be quite large, raising the need to exploit the knowledge of what happened in-between the observations.

There is very little literature about how to estimate the parameters of a discrete route choice model with proximity data. This in contrast with route choice model estimation using location data, for which plenty of estimation-related methods have been proposed, ranging from map matching and trajectory reconstruction algorithms, in order to augment incomplete data, to more elaborate methods, like the network-free data estimation approach introduced by Bierlaire and Frejinger (2008). In this paper, we therefore focus explicitly on estimation of a discrete route choice model with proximity data. We describe and apply an estimation method that exploits the very specific nature of this type of data with respect to location data. The framework in which we describe the methodology allows us to formulate a probabilistic link utilization estimation method as well, which can be used to derive link flows and route splits from a set of sensor observation sequences. We implemented and tested this method as well.

The remainder of this paper is organized as follows. Section 2 briefly reviews the different estimation techniques of route choice models, making the explicit distinction between using location data sources and proximity data sources. Then, Section 3 describes the framework that encapsulates our estimation method. This section comprises a brief description of the adopted Recursive link-based Logit model and a precise formulation of the sensor network configuration. Section 4 describes how observations of individuals travelling through the network are represented in terms of sensor observations, and in Section 5 we focus on the particular case of trips without any observations. Insights into unobserved travelling are generalized in Section 6, in which a method is derived to calculate the likelihood to reproduce any sequence of sensor observations. This likelihood calculation is the key element of the route choice model estimation method, which is applied in Section 7 on an artificial network and randomly generated observation patterns. Section 8 describes a probabilistic link utilization estimation method, which is based on similar principles as the likelihood calculation method. In Section 9, this method is tested in a simulated use-case. Then, Section 10 describes how we applied the developed methodologies on a Wi-Fi data set that we collected during the TT Assen, a music festival in the Dutch city Assen. Section 11 compares our model estimation method with two different implementations of the network-free data approach (Bierlaire and Frejinger, 2008). We show that with a particular implementation of the measurement equation, the network-free data approach can be interpreted as a path-based version of our recursive method. Section 12 discusses the computational complexity of the model estimation method and Section 13 discusses the applicability of the model in different contexts. The paper ends with conclusions and future steps in Section 14.

2. Review of route choice model estimation approaches

Many types of data have been used to estimate the parameters of discrete route choice models, or traveller behavior models in general. Traditionally, surveys were commonly used to estimate model parameters, but with the rise of new digital technologies, estimation shifted more towards the use of behavioral observations from sensors. In general, data sources to estimate route choice or location choice models can be classified into two groups; data sources reporting locations (location data sources) and data sources reporting proximities to certain (in most cases fixed) nodes (proximity data sources). In this paper, we treat proximity as a binary concept; either, the individual is close to the node, or he is not. We will now briefly review both data source classes, with a focus on its use in route choice model estimation.

2.1. Location data sources

Data sources that directly report locations of individuals mainly include self-reported behavior (Mahmassani and Peeta, 1993; Abdel-Aty et al., 1995; Ramming, 2001) and GPS measurements (Broach et al., 2012; Menghini et al., 2010; Ton et al., 2018; Galama, 2015). Both types of data contain errors and uncertainties. In the case of collected GPS-traces or reported routes, data might not correctly represent the true routes of people. It is known that the accuracy of GPS-measurements is highly influenced by the environment and the high battery consumption of GPS-localization limits the maximum frequency. Moreover, the limited ability of humans to reproduce their taken routes makes reported route data also unreliable to a certain extent. To deal with the problem of incomplete trajectory data, reconstruction of trajectories seems a logical choice, frequently by assuming a shortest path choice (Ramming, 2001; Lu et al., 2018). Up to certain levels of inaccuracy and gap duration, this might work fine and can give satisfying results. Bierlaire and Frejinger (2008) however warn that biases are easily introduced when applying these trajectory reconstruction techniques. As an alternative, he proposes a method to estimate route choice models with network-free data, reducing the need for trajectory reconstruction and map-matching.

(a) Location data source

(b) Proximity data source

Fig. 1. Example showing the difference between (a) location data sources and (b) proximity data sources, with respect to route likelihood. In the case of location data it is impossible to distinguish from the two observations whether the individual took the upper or the lower (actual) route. In the case of proximity data, the absence of an observation of the individual near the sensor at the upper node makes the lower route much more likely to be the actual one.

Oyama and Hato (2018) propose a network-free estimation method which relies on a link-based route choice model and reduces biases by estimating link-specific standard errors. Both methods use location data to estimate the model parameters.

2.2. Proximity data sources

Data sources that report when individuals are close to certain nodes include technologies as Wi-Fi-sensing, Bluetooth, RFID, or mobile phone network data. Wi-Fi-traces have been used to study activity and destination choice by inferring trip origins and destinations from the traces (Danalet et al., 2014; Danalet, 2015; Yoon et al., 2006). Yoon et al. (2006) studied route choice behavior as well, by generating a distance-based set of path alternatives, and statistically deriving route splits from the data. Also mobile phone network data has been used to study destination choice (Iqbal et al., 2014; Wang et al., 2018) as well as route choice (Leontiadis et al., 2014; Huang et al., 2018). Huang et al. (2018) estimated the perception parameter of a C-logit model with so-called antenna ID paths. van den Heuvel et al. (2015) used Bluetooth scan-units to estimate a route choice model in a train station. The scan-units were placed such that all alternative routes could be unambiguously observed. In urban networks, achieving full observability of all possible routes is generally infeasible. In these contexts, Bluetooth observations from selected locations in the network are often used to approximate densities, flows and travel or waiting times (Versichele et al., 2012; Larsen et al., 2013; Kurkcu and Ozbay, 2017; Lesani and Miranda-moreno, 2018). However, to the best of our knowledge, a general and dedicated estimation method for a discrete route choice model has not been developed for this type of data sources so far.

2.3. Key difference between location and proximity data

As briefly explained in the introduction (Section 1), there is a fundamental difference when we compare location data and proximity data when it comes to route choice model estimation. For location data sources, only actual measurements provide information about the actual taken route. Existing trajectory reconstruction and estimation techniques are generally based on the measured locations of an individual only. In contrary, for proximity data sources, also the absence of sensor observations contributes to the likelihood of routes that do not cross the particular sensor. This key difference is visualized by Fig. 1, in which the absence of an observation of the individual near a sensor at the upper node (Fig. 1(b)) makes the lower route more likely to be the actual one. This particular fact makes existing route estimation methods as proposed by Bierlaire and Frejinger (2008) and Oyama and Hato (2018) not applicable, or at least not optimal, for estimation of route choice models using proximity data. The method proposed in this paper does explicitly take the absence of observations into account, by calculating likelihoods for individuals to exactly reproduce the sequence of sensor observations, herewith avoiding those places where they have not been observed. Before explaining the estimation method in detail, the framework in which the method will be integrated will be outlined in the next section.

3. Modeling framework

A link-based network representation will be used for both our sensor configuration and the route choice model. Before formally describing the sensor configuration in Section 3.1 and the route choice model in Section 3.2, we start with some general network and route definitions.

We introduce a network $G = (\mathcal{L}, \mathcal{V})$, with directed links $\mathcal{L}$ and nodes $\mathcal{V}$. A link $l \in \mathcal{L}$ is defined to have a start and end vertex: $l = (v_1, v_2)$, with v_1 and v_2 both in $\mathcal{V}$. A path r through the network is defined as a sequence of links $(r_1, r_2, \ldots)$, with $r_i \in \mathcal{L}$ for all i.

Given the destination and current link of an individual, we assume that the probabilities of choosing a next link are known. Different methods exist to define these probabilities. In this study, a Recursive link-based Logit (RL) model has been applied, which is formulated in terms of a next-link probability matrix. Section 3.2 briefly explains how this model is used to determine the next-link probabilities. Regardless of the exact model, we define $p_{i,j,d} = P(j|i,d)$ as the probability of choosing link $j \in \mathcal{L}$ as the next link when located at link $i \in \mathcal{L}$ and having link $d \in \mathcal{L}$ as destination. We assume that the next-link

choice does not depend on the historical path. If the destination link is reached, a person is expected to stop moving, so $p_{d,j,d} = 0$ for all d and j in $\mathcal{L}$. Furthermore, a person that arrives at the start node of the destination link is expected to choose the destination link as its next (and final) link, so $p_{i,d,d} = 1$ for all links i that can precede destination d.

Later on, it is more convenient to write these probabilities in matrix notation, so we introduce the next-link probability matrix $\mathbf{P}_d$ with entries $(\mathbf{P}_d)_{i,j} = p_{i,j,d}$. Although we are formally not allowed to use link elements as matrix indices, for the benefit of a clear notation, we implicitly assume an ordering of all links, which we use for indexing.

3.1. Sensor configuration

In order to describe a sensor configuration, we first introduce $\mathcal{S}$ as being the set of all sensors. Then, we model the sensor configuration by matching each sensor $s \in \mathcal{S}$ with a non-empty set of one or more links to which an observation of that sensor possibly applies. We denote the relation between a sensor and its corresponding set of links by the function $\mathcal{L}_S$, such that $\mathcal{L}_S(s)$ equals the set of links that are observed by sensor s. The set of all links that are observed by a sensor is denoted by $\mathcal{L}^* = \bigcup_{s\in\mathcal{S}} \mathcal{L}_S(s)$.

The estimation methods, described in the next sections, put two important requirements on the construction of the observed link sets. First, each link is allowed to be in the observed link set of at most one sensor. This restriction largely simplifies the estimation methods, but poses a restriction on the application scope as well (see Section 13). Second, the observed link set of a sensor should be constructed in such a way that each possible non-cyclic path that crosses the detection area of the sensor should have exactly one link that is in the observed link set. The second requirement comes from the fact that we actually aim to calculate the likelihood to reproduce the observed sensor crossings, instead of the likelihood to reproduce the exact observed sensor observations. This simplifies the derivation of our methodology and, in addition, it has a positive effect on the computational efficiency.

Modelling the observed area of a sensor as a set of one or more observed links allows for many different configurations. Four typical ways to construct the observed link set are:

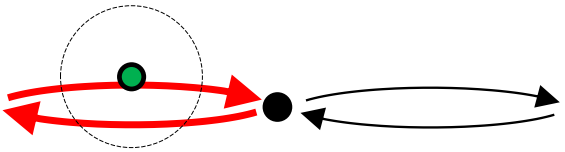
- **Single-link construction**: In this construction, a sensor simply observes one (possibly bi-directional) link. Practically it implies that an individual that is observed by this sensor undoubtedly traverses this link. See Fig. 2(a) for an example.
- **Single-node construction**: In this construction, a sensor observes one single node (intersection). Since each possible path through the node should have exactly one link that is in the observed set (second requirement), the observed link set is constructed from all incoming links. See Fig. 2(b) for an example.
- **Multi-node construction**: In this construction, a sensor observes multiple nodes. Since each possible path crossing the detection area should have exactly one link that is in the observed set (second requirement), the observed link set is constructed from all links that enter the detection area. See Fig. 2(c) for an example.
- **Dummy-link construction**: This construction is similar to the single-node and multi-node construction, with the key difference that dummy-links are inserted that connect incoming and outgoing links. The observed link set then consists of all dummy-links. See Fig. 2(d) for an example.

It is known for passive Wi-Fi- or Bluetooth-sensors that the detection rate is far from 100%. Therefore, we assume that each link l is associated with a link-specific detection probability θ_l, with $0 \leq \theta_l < 1$. Obviously, $\theta_l = 0$ for all unobserved links ($l \notin \mathcal{L}^*$). In practice, the detection rate θ_l will depend mainly on the utilized sensor and its placement with respect to the local surroundings and infrastructure. Generally, the detection rate is expected to increase with a longer duration of stay and shorter distances from the sensor. This implies that the detection rate could change if a person chooses a different path in the detectable area (e.g., making a turn instead of crossing the street). In case the detection rate by a *single-node* or *multi-node* sensor depends heavily on the exact path of the individual crossing the detectable area, a *dummy-link* construction could be considered. This would allow for a more direct specification of detection rates for different paths through the detectable area.

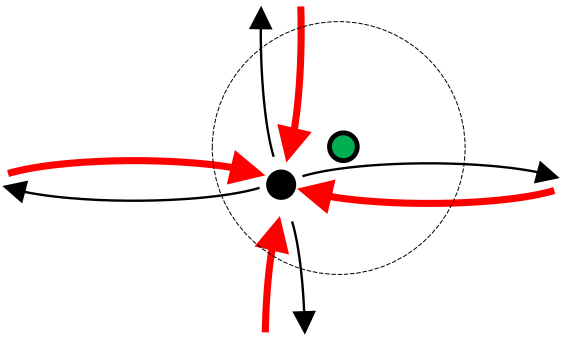
3.2. The recursive link-based logit model

This section briefly reviews the Recursive link-based Logit model, which is used in this study to define the next-link probabilities of individuals in a network that travel towards a destination. The section does not contain any new ideas or insights, although the notation differs slightly from the notation used by Fosgerau et al. (2013). The Recursive link-based Logit model (RL) was introduced by Fosgerau et al. (2013) as an alternative to existing discrete choice models to describe route choice behavior. The main advantage of the RL model is that it has no restriction on the choice set. Its specifications are comparable to existing traditional discrete choice methods. In the RL model, when currently at link i, the action of choosing a next link j has an instantaneous utility $v(j|i) + \mu\epsilon(j)$, where the stochastic $\epsilon(j)$ terms are assumed i.i.d. extreme value type 1 with zero mean and μ is a fixed scale parameter. A person travelling from link i to destination link d is modelled to maximize its total expected accumulated utility. The expected accumulated utility can be found by solving the Bellman equation:

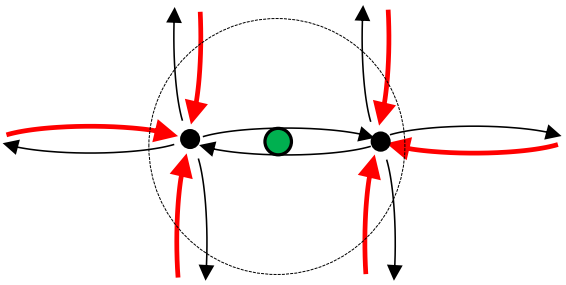
$$V_{i,d} = E\left[\max_{j\in\mathcal{L}(i)} \left(v(j|i) + V_{j,d} + \mu\epsilon(j)\right)\right], \tag{1}$$



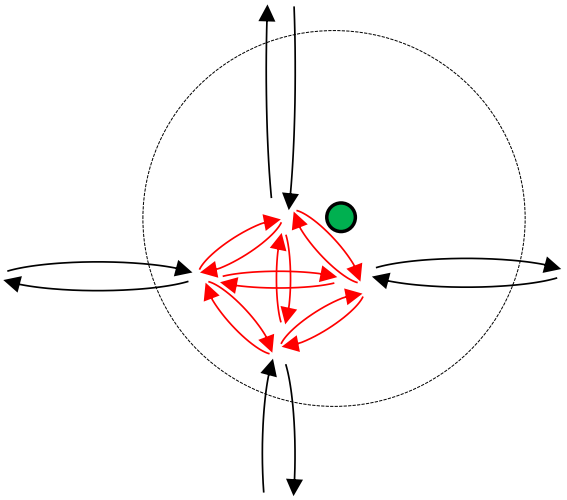
(a) Single-link construction: sensor observes one (bi-directional) link.



(b) Single-node construction: sensor observes incoming links of a node.



(c) Multi-node construction: sensor observes all incoming links of the detection area that covers multiple nodes.



(d) Dummy-link construction: sensor observes dummy links that connect the incoming and outgoing links of the detection area.

Fig. 2. Four constructions for the observed link set $\mathcal{L}_S(s)$ of sensor s. The sensor location and its detection range are indicated by the green solid circle and the dashed outlined circle respectively. The links in $\mathcal{L}_S(s)$ are indicated in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where $\mathcal{L}(i)$ represents the choice set of next links when currently travelling at link i. Since the error terms are assumed i.i.d. extreme value type 1 and μ is invariant, the equation can be rewritten as

$$V_{i,d} = \begin{cases} \mu \ln\left(\sum_{j\in\mathcal{L}(i)} e^{\frac{1}{\mu}(v(j|i)+V_{j,d})}\right), & i \neq d \\ 0, & i = d \end{cases} \tag{2}$$

(See Fosgerau et al. (2013)).

Taking the exponential of both sides of the equation leaves us with a linear system of equations in terms of the accumulated utility exponentials. Therefore, we define an incidence matrix of instantaneous utilities $\mathbf{M}_d$ with entries according to

$$(\mathbf{M}_d)_{i,j} = \delta(j|i)e^{\frac{1}{\mu}v(j|i)}, \tag{3}$$

where $\delta(j|i)$ is 1 if link j is a neighbour of link i and zero otherwise. $\delta(j|d) = 0$ for all j, since the person is expected to stop moving when arriving at its destination link. Further, we define the vector $\mathbf{b}$, which has $b_d = 1$ and $b_i = 0$ for $i \neq d$, and the vector $\mathbf{z}_d$, for which $(\mathbf{z}_d)_i = e^{\frac{1}{\mu}V_{i,d}}$.

The accumulated utility exponentials $\mathbf{z}_d$ can now be found by solving

$$(\mathbf{I} - \mathbf{M}_d)\mathbf{z}_d = \mathbf{b}. \tag{4}$$

The next-link probability matrix $\mathbf{P}_d$ is finally determined by

$$(\mathbf{P}_d)_{i,j} = \frac{(\mathbf{M}_d)_{i,j}(\mathbf{z}_d)_j}{\sum_{l\in\mathcal{L}}(\mathbf{M}_d)_{i,l}(\mathbf{z}_d)_l} \tag{5}$$

Link flows that result from a given demand $\mathbf{g}_{o,d}$ between origin o and destination d can now be calculated according to

$$\mathbf{q}_{o,d} = (I - \mathbf{P}_d^T)^{-1}\mathbf{g}_{o,d}. \tag{6}$$

In this formula, the vector $\mathbf{q}_{o,d}$ contains the link flows and $\mathbf{g}_{o,d}$ is a vector that is zero except for its oth element, which contains the demand from o to d.

3.3. Link size attribute

The original formulation of the RL model suffers from the IIA (independence in alternatives) property. In order to relax this property, Fosgerau et al. (2013) proposed a link size attribute, which is comparable to the path size attribute in path choice models (Ben-Akiva and Bierlaire, 1999). The correction is achieved by adding a term to the instantaneous link-transition utilities, which is proportional to the flow through this link resulting from a unit of demand between an origin and a destination. The corrected instantaneous utility is defined as

$$v_{LS}(j|i,o,d) = v(j|i) + \beta_{LS} \cdot (\tilde{\mathbf{q}}_{o,d})_j \cdot \Delta(j), \tag{7}$$

where $\Delta(j)$ denotes the length of link j, β_{LS} denotes the link size parameter and $(\tilde{\mathbf{q}}_{o,d})_j$ denotes the flow through link j resulting from a unit of demand between origin o and destination d. The factor β_{LS} is supposed to be negative. This implies that links with a large flow, which are likely to have a large contribution to route overlap, get a larger utility reduction. The flow vector $\tilde{\mathbf{q}}_{o,d}$ is calculated according to Eq. (6) and requires a certain choice of $\mathbf{P}_d$, denoted as $\mathbf{P}_{corr,d}$. Fosgerau et al. do not prescribe how to choose the matrix $\mathbf{P}_{corr,d}$ in order to calculate the link size correction flows. A typical choice would be to derive $\mathbf{P}_{corr,d}$ from assuming a utility function that is based on trip distance only (and u-turn penalties). It should be noticed that the link size utility function $v_{LS}(j|i,o,d)$ is origin-specific. This makes that the next-link probability matrix $\mathbf{P}_d$ and the incidence matrix $\mathbf{M}_d$ become origin-specific as well, which enlarges the computational effort to estimate models. In our study, the link size attribute has been applied when estimating the model for the case-study at the TT music festival (Section 10). In the other parts of the paper, a link size attribute has not been taken into account, mainly to ensure readability. Although details are omitted, the estimation method and simulations that will be introduced in the following sections trivially allow for inclusion of the link size attribute.

A more profound approach to relax the IIA property is to use the Nested Recursive Logit (NRL) model (Mai et al., 2015; Zimmermann et al., 2017), which explicitly accounts for correlated path utilities. Although the NRL model has proven to be superior to the RL model in terms of its accuracy, its higher complexity has made us decide to take the RL model as the starting point for our method development.

4. Sensor observations

Since the full paths of persons generally cannot be observed, we define for each individual a sensor observation path $s^* = (s_1^*, s_2^*, \ldots)$ as the sequence of sensors at which a person during a trip has been observed. The set of all observation paths for trips with origin link o and destination link d is denoted by $S_{o,d}^*$.

It should be noticed that timestamps of the observations are deliberately not taken into account. Inclusion of the time dimension complicates the likelihood calculation as explained in the coming sections and has therefore been left out. In case of estimating a route choice model for pedestrians in an urban context or during an event, which this method is particular aimed at, it is questionable how to deal with this time aspect, since pedestrians in these environments are not expected to move with predictable speeds. Nonetheless, how to include timestamps into our likelihood estimations in order to improve the predictive power is one of our key questions to be answered by future research.

5. Unobserved travelling

Before we explain how to calculate sensor observation path likelihoods and link flows, we will briefly discuss the concept of unobserved travelling. A quantity that appears to be crucial in our later computations is the probability of passing a certain link i, when travelling from a certain origin to a certain destination, and being unobserved so far. In other words, the individual has not been observed yet by a sensor before reaching link i. To make this formal, we introduce $q^0(i|o,d)$ as being the expected number of times that an individual *arrives* unobserved at link i, when travelling from o to d:

$$q^0(i|o,d) = \mathbb{E}(k^0(i)|o,d), \tag{8}$$

where $k^0(i)$ equals the number of times that an individual arrives unobserved at link i. This quantity depends on the amount of routes between the origin and destination that pass through link i and on the detection rates of the sensors that an individual passes before reaching link i. Fig. 3 shows two examples of how the network and sensor configuration influence $q^0(i|o,d)$. It should be noticed that the expected number of unobserved link arrivals is almost identical to the probability of arriving unobserved at the specific link at least once. The small, but non-zero, probability of cycles to occur, makes the expected number of unobserved link arrivals slightly larger than the probability of arriving unobserved at the specific link at least once.

(a) (b)

Fig. 3. The expected number of unobserved link arrivals, $q^0(i|o,\ d)$, for two different sensor configurations (*single-node* construction) with a detection rate $\theta = 0.7$. The wider and greener the link, the higher is the expected number of unobserved link arrivals. The most left link is defined as the origin and the most right link as the destination. The next-link probability matrix $\mathbf{P}_d$ is based on link distances only plus a u-turn penalty. It can be seen that $q^0(i|o,\ d)$ is practically zero for the origin link, since only link arrivals are counted.

The values of $q^0(i|o,\ d)$ are found by calculating 'link flows' that result from sending one unit of flow into the network at the origin link o, with a modified next-link probability matrix, which takes the link-specific detection probabilities into account. Each flow that passes a sensor-equipped link will be lowered according to its detection rate θ. Details are given in Appendix A.

6. Likelihood of sensor observations

This section explains how to calculate the joint likelihood $L(\boldsymbol{\beta})$ of reproducing the set of sensor observation paths $S^*_{o,d}$ for all o-d pairs, given a parameter set $\boldsymbol{\beta}$, which influences the element values of the next-link probability matrix or detection rates. This likelihood can be maximized in order to estimate utility parameters (Ben-Akiva and Lerman, 1985). The joint likelihood is calculated by multiplying all probabilities to observe the individual paths:

$$L(\boldsymbol{\beta}) = \prod_{o\in\mathcal{L}} \prod_{d\in\mathcal{L}} \prod_{\substack{s^*\in \\ S^*_{o,d}}} P(s^*|o, d, \boldsymbol{\beta}), \tag{9}$$

where $P(s^*|o,\ d,\ \boldsymbol{\beta})$ is the likelihood to reproduce the sensor observation path s^*, given origin o, destination d and parameter set $\boldsymbol{\beta}$. To calculate $P(s^*|o,\ d,\ \boldsymbol{\beta})$, we distinguish between empty and non-empty sensor observation paths. The following sub sections describe how to find the likelihoods for both categories. For simplicity, from now on we will omit the $\boldsymbol{\beta}$ term in our notation, since all further derivations do not explicitly depend on $\boldsymbol{\beta}$.

6.1. Likelihood of empty observation path

Given an origin link o and a destination link d, $P(\emptyset|o,\ d)$ is the probability that an individual that travels from o to d is not observed by a single sensor. In order to calculate the likelihood of an empty sensor observation path, we use the expected number of unobserved arrivals at the destination link, $q^0(d|o,\ d)$ (see Section 5). It should be noticed that a person can only reach the destination once, since $p_{d,j,d}$ is defined to be 0 for each destination d and link j. As a result, $q^0(d|o,\ d)$ equals the probability to arrive at the destination link unobserved, which allows us to write:

$$P(\emptyset|o, d) = \begin{cases} 1, & \text{if } o = d \\ (1-\theta_d)\cdot q^0(d|o, d), & \text{otherwise.} \end{cases} \tag{10}$$

Eq. (10) simply states that the likelihood of an empty sensor observation path equals the probability to arrive unobserved at the destination link, multiplied with the non-detection rate of the destination link. In case the origin and destination are the same, the likelihood obviously equals 1.

6.2. Likelihood of non-empty observation path

Given an origin link o and a destination link d, $P((s^*_1, s^*_2, \ldots, s^*_n)|o, d)$ is the probability that an individual that travels from o to d is observed by the sensors $(s^*_1, s^*_2, \ldots, s^*_n)$, in the given order. Since the first sensor observation s^*_1 is associated with exactly one link in the link set $\mathcal{L}_S(s^*_1)$, we may express this probability as a simple sum:

$$P((s^*_1, s^*_2, \ldots, s^*_n)|o, d) = \sum_{l\in\mathcal{L}_S(s^*_1)} P((l, s^*_2, \ldots, s^*_n)|o, d), \tag{11}$$

where $P((l, s^*_2, \ldots, s^*_n)|o, d)$ denotes the probability to be first observed at link l, followed by the sensors s^*_2 to s^*_n. Since actual choices are assumed to be independent of historical choices, this probability can be decomposed as follows:

$$P((l, s^*_2, \ldots, s^*_n)|o, d) = q^0(l|o, d)\cdot\theta_l\cdot P((s^*_2, \ldots, s^*_n)|l, d) \tag{12}$$

The term $q^0(l|o,\ d)\cdot\theta_l$, the expected number of unobserved link l arrivals times the detection rate, equals the probability that an individuals first observation happens at link l. The term $P((s^*_2, \ldots, s^*_n)|l, d)$ equals the probability to reproduce the

remaining sensor observation path, starting at link l. Substituting Eq. (12) into Eq. (11) gives us the following recursive scheme:

$$P((s_1^*, s_2^*, \ldots, s_n^*)|o, d) = \sum_{l \in \mathcal{L}_S(s_1^*)} q^0(l|o, d) \cdot \theta_l \cdot P((s_2^*, \ldots, s_n^*)|l, d). \tag{13}$$

With this equation, the likelihoods of the observation paths can be calculated recursively using a standard dynamic programming top-down approach, in which earlier results of (sub-)problems are stored and re-used, which is called memoization (Cormen et al., 2009). At first sight, it could appear that the computational complexity of the calculation can blow-up very easily in case of long observation paths and many observed links per sensor. However, we notice that calculation of $P((s_2^*, \ldots, s_n^*)|l, d)$ involves calculation of $P((s_3^*, \ldots, s_n^*)|l_2, d)$ for all $l_2 \in \mathcal{L}_S(s_2^*)$, which is regardless of the link l. The fact that the sub-probabilities to be calculated are independent of the followed path in the recursion tree, makes that the number of function evaluations does not grow exponentially. See Section 12 for a comprehensive examination of the computational complexity.

In Section 7, we use the likelihood calculation to estimate the parameters of a RL-model with an artifical data set. In Section 10, the method is applied on a real data set that was collected during the TT Assen Festival.

7. Simulated use-case: Estimating an RL model

To analyze the applicability of the likelihood calculation (see Section 6), we will now use it to estimate a Recursive Logit (RL) model. This section describes the methodology to estimate the parameters based on a simulated data set of agents that move through partially observed networks. To evaluate the performance of the methodology, we first generate a number of agent paths through the network, according to an RL model with predefined parameter values $\boldsymbol{\beta}_0$, which serves as a ground truth. These network paths are then reduced to sensor observation paths by checking which links in a path are covered by which sensors and taking the detection rates into account. Then, we look for those RL parameter values $\boldsymbol{\beta}_{est}$ that maximize the log-likelihood of these sensor observation paths. Finally, we compare the estimated parameter values $\boldsymbol{\beta}_{est}$ with the original values $\boldsymbol{\beta}_0$, as well as the resulting network use of agents.

7.1. Network and behavior

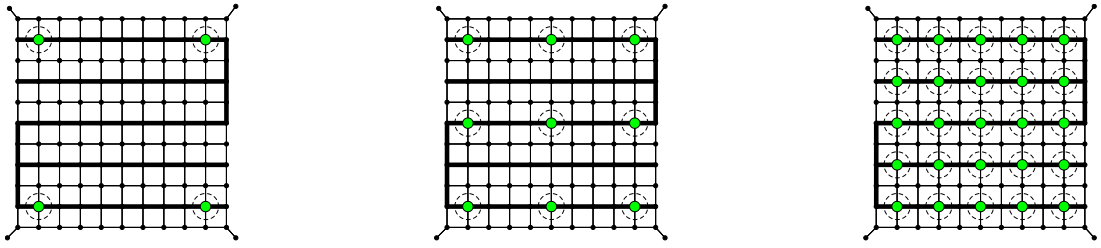
A network has been defined that consists of 224 bi-directional links from which four are defined as an origin/destination link. See Fig. 4 for a schematic visualization of the network. The origin/destination links are diagonally connected to the corners of the network.

Since we want our model to estimate preferences regarding different link characteristics, we defined two different road types, which are indicated in the figure by their line thickness. Three different sensor configurations have been tested, shown in Fig. 4(a), (b) and (c). Sensors are placed at nodes (*single-node* construction, Fig. 2(b)) and are visualized as large green dots. Each sensor has a detection rate θ.

We define the following instantaneous utilities to move from link i to link j:

$$v(j|i) = -(1 + \beta_{R1} \cdot \mathbf{1}_{R_1}(j) + \beta_{R2} \cdot \mathbf{1}_{R_2}(j)) \cdot \Delta(j) - c_{penalty} \cdot \mathbf{1}_U(i, j). \tag{14}$$

In this expression, $\mathbf{1}_{R_1}(j)$ is an indicator function which evaluates to 1 in case link j belongs to the set of links with road type 1 (thin line) and 0 otherwise. The function $\mathbf{1}_{R_2}(j)$ evaluates to 1 in case link j belongs to the set of links with road type 2 (thick line), 0 otherwise. The function $\Delta(j)$ denotes the length of link j. β_{R1} and β_{R2} are the road type-specific utility parameters. If both parameters would be 0, the utility would be determined by route length only. Finally, the function $\mathbf{1}_U(i, j)$ evaluates to 1 in case the transition from link i to link j is a u-turn and 0 otherwise. Multiplied with a fixed constant



(a) 4 sensors (b) 9 sensors (c) 25 sensors

Fig. 4. The network used for the simulated use-case. The thickness of the line represents the road type (thin line = type 1, thick line = type 2). Three different sensor configurations (shown in sub figures a, b and c) with an increasing number of sensors, which are indicated by the large green dots, have been tested. The four diagonal links connected to the corners are origin/destination links and are modelled to have zero length. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$c_{penalty}$, which will not be varied during the optimization, and subtracted from the distance term, this effectively prevents the agent from making u-turns.

For each OD-pair, a total of n paths have been generated according to the RL model with utilities as in (14), a scale parameter μ (fixed to 1) and road type-specific utility parameters $\boldsymbol{\beta}_0 = (\beta_{R1,0}, \beta_{R2,0})^T$. From these travelled paths, sets of OD-specific sensor observation paths have been generated, $S^*_{o,d}$.

7.2. Log-likelihood optimization

The joint log-likelihood of the generated sensor observation paths has been maximized by varying the 'to-be-estimated' parameters $\boldsymbol{\beta} = (\beta_{R1}, \beta_{R2})^T$:

$$\boldsymbol{\beta}_{est} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \ln\left(L(\boldsymbol{\beta})\right) \tag{15}$$

$$= \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \sum_{o\in\mathcal{L}} \sum_{d\in\mathcal{L}} \sum_{s^*\in S^*_{o,d}} \ln\left(P(s^*|o,d,\boldsymbol{\beta})\right) \tag{16}$$

The optimization was performed using MATLAB's non-linear constrained optimizer (function *fmincon*). Gradients were approximated with finite differences. A lower bound of -0.99 was set as a constraint for both parameter values, because as soon as one of the parameter values drops below -1, link utilities may become positive, causing a potential preference for infinite trip lengths.

7.3. Evaluation

The estimated model was evaluated using a series of metrics.

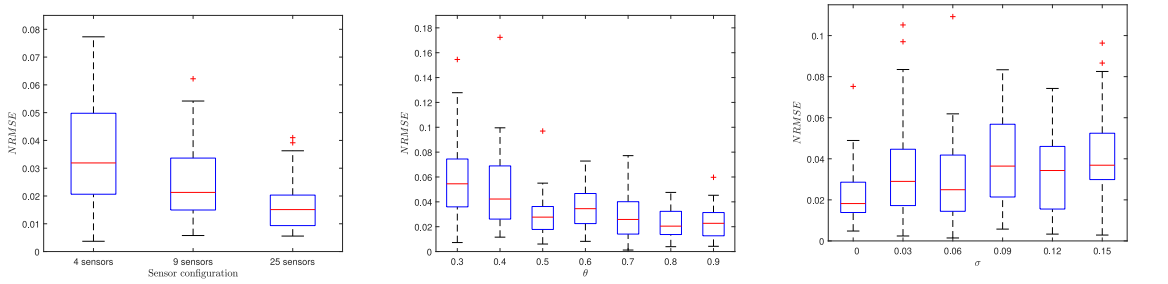
- $t_{stat,R1}$ and $t_{stat,R2}$. These variables denote the t-statistics for the parameters $\beta_{R1,est}$ and $\beta_{R2,est}$ respectively. Their values are calculated by dividing the parameter values $\beta_{R1,est}$ and $\beta_{R2,est}$ by their standard errors, which we estimated by the Cramér–Rao lower bound. The Hessian of the log-likelihood, which is involved in this calculation, was approximated by finite differences.
- ρ^2. This metric is defined as $\rho^2 = 1 - \frac{\ln\left(L(\boldsymbol{\beta}_{est})\right)}{\ln\left(L(\boldsymbol{\beta_0})\right)}$ and is a measure for the model fit. In this example, $\boldsymbol{\beta_0} = \mathbf{0}$. It should be noticed that the log-likelihoods in this formula apply to the sensor observation paths, and not directly to route choice behavior itself. Therefore, the value of ρ^2 should be interpreted with care and not be directly compared with ρ^2 values that are calculated from the complete route perspective.
- *RMSE*. In order to make a statement about the predictive performance with respect to real network use, we calculate the root mean square error of weighted link flows: $RMSE = \sqrt{\sum_{l\in\mathcal{L}} w_l \cdot \left(\tilde{\mathbf{q}}_{est,l} - \tilde{\mathbf{q}}_{0,l}\right)^2}$. In this formula, $\tilde{\mathbf{q}}_{est,l}$ and $\tilde{\mathbf{q}}_{0,l}$ are the link flows, in the estimated and simulated case respectively, that result from a demand of 1 for each OD-pair (according to Eq. (6)). The mean is weighted according to link length: $w_l = \Delta(l) / \sum_{i\in\mathcal{L}} \Delta(i)$, where $\Delta(l)$ denotes the length of link l.
- *NRMSE*. The normalized root mean square error of weighted link flows $NRMSE = RMSE / \sum_{l\in\mathcal{L}} w_l \cdot \tilde{\mathbf{q}}_{0,l}$. Since the *NRMSE* is normalized by the mean link flow, its value is expected to be less depending on the network size and number of OD-pairs.

For 7 different sets of parameters, we performed 30 simulations, for which we all estimated model parameters $\beta_{R1,est}$ and $\beta_{R2,est}$. For each parameter set, the average and standard deviation of the estimated parameters and evaluation metrics are reported in table 1.

Table 1 shows that the method rather precisely rediscovers the parameter values that were used to generate the data sets. The t-statistics indicate that the values are also significantly different from 0, unless they were supposed to be 0. Also with respect to the predictive performance, the method seems to perform really well, since the average *NRMSE* over 30 simulations does not exceed 5%.

Table 1
Estimated parameters $\beta_{R1,est}$ and $\beta_{R2,est}$ and evaluation metrics for three different parameter sets. The values report the mean and standard deviation over 30 simulations. Default parameter values: $n = 100$, $\mu = 1$, $c_{penalty} = -100$, $\theta = 0.7$.

#sensors	$\beta_{R1,0}$	$\beta_{R2,0}$	$\beta_{R1,est}$	$\beta_{R2,est}$	$t_{stat,R1}$	$t_{stat,R2}$	ρ^2	RMSE	NRMSE
4	0.00	0.50	−0.00 ± 0.05	0.52 ± 0.06	−0.05 ± 1.1	8.6 ± 0.22	0.17 ± 0.01	1.4 ± 0.82	0.04 ± 0.02
9	0.00	0.00	0.01 ± 0.02	0.00 ± 0.02	0.37 ± 1.1	0.12 ± 0.96	0.00 ± 0.00	1.2 ± 0.47	0.03 ± 0.01
9	0.00	0.50	−0.00 ± 0.03	0.50 ± 0.05	−0.05 ± 1.0	11 ± 0.17	0.15 ± 0.01	1.0 ± 0.58	0.03 ± 0.01
9	0.50	0.00	0.50 ± 0.04	0.00 ± 0.04	12 ± 0.33	0.03 ± 1.00	0.26 ± 0.01	0.83 ± 0.61	0.02 ± 0.02
9	−0.10	0.30	−0.10 ± 0.03	0.31 ± 0.03	−3.87 ± 0.97	9.8 ± 0.50	0.11 ± 0.01	1.3 ± 0.79	0.03 ± 0.02
9	0.10	−0.30	0.10 ± 0.02	−0.30 ± 0.02	4.2 ± 0.92	−12.73 ± 1.3	0.16 ± 0.01	1.3 ± 0.55	0.03 ± 0.01
25	0.00	0.50	−0.00 ± 0.02	0.50 ± 0.03	−0.18 ± 0.95	16 ± 0.21	0.12 ± 0.01	0.67 ± 0.39	0.02 ± 0.01



(a) *NRMSE* versus number of sensors

(b) *NRMSE* versus detection rate θ

(c) *NRMSE* versus standard deviation σ of actual detection rates

Fig. 5. The *NRMSE* for (a) different sensor configurations, (b) different detection rates θ and (c) different levels of uncertainty of actual detection rates. All figures show a box plot, showing the median (red line) and the 25th and 75th percentiles (blue edges of box). Whiskers of the box plot extend to the most extreme points that are not considered outliers. The outliers are plotted individually using the '+' symbol. Each box plot is based on 30 simulation runs. Default parameter values: $n = 100$, $\beta_{R1,0} = 0$, $\beta_{R2,0} = 0.5$, $\mu = 1$, $c_{penalty} = -100$, #sensors = 9 (Fig. 4(b)) and $\theta = 0.7$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

7.4. Relation between predictive performance and sensor characteristics

To get a better understanding of the relation between the sensor configuration and the predictive performance of the estimated model, three analyses have been performed.

First, we looked at the effect that the number of sensors has on the *NRMSE*. The result of testing the three different sensor configurations (Fig. 4(a), (b) and (c)) is shown in Fig. 5(a). With more sensors, the predictive performance of the model increases (*NRMSE* decreases), which is in line with our expectations. Clearly, the position of sensors is an important factor as well, but a thorough analysis of the effects of sensor positions will be left for future research.

Second, we looked into the effect of the detection rate θ on the *NRMSE*. For detection rates between 0.3 and 0.9, we calculated the *NRMSE* (mean and standard deviation) over 30 simulations. The results are shown in Fig. 5(b). The figure shows a general increase of the predictive performance (decreasing *NRMSE*) with increasing detection rate. The apparent local minimum around $\theta = 0.5$ was not present in a second run (from which we did not show results in this paper), so we attribute the presence of this minimum to the stochastic nature of our assessment.

Third, we studied the predictive performance in case of uncertainty about the actual sensor detection rates. So far, we assumed detection rates to be deterministic. For this analysis, during sensor observation paths generation, the actual sensor detection rates were randomly drawn from a Gaussian distribution with a mean of 0.7 and a standard deviation σ. During parameter estimation, we assumed the detection rates' deviations from their means to be unknown, so all rates were considered to be equal to $\theta = 0.7$. The relation between the resulting *NRMSE* and σ gives an idea about the effect of the detection rate uncertainty with respect to predictive performance. Fig. 5(c) shows the results. As expected, the predictive performance decreases with increasing uncertainty, which shows the need for a proper understanding of our sensor detection characteristics. At the same time, we conclude that introduction of the detection uncertainty does not drastically lower the predictive performance.

8. Link utilization estimation from sensor observation paths

Besides calculating the likelihoods of a given sensor observation path, we can estimate the link utilization from sensor observation paths. Without any sensor information, our best stochastic guess would be that the route for an individual moving from origin o to destination d would be described by the link flows as already calculated by Fosgerau (see Eq. (6)). However, knowing at which locations the individual was identified and where he or she was not identified, we can improve these link utilization estimations. For this end, we follow a similar approach as for the sensor observation path likelihood calculations (Section 6).

First of all, we have to define link utilization as being conditional with respect to a measured sensor observation path. Therefore, we introduce $q(i|o, d, s^*)$, as being the expected number of times that link i is visited given the sensor observation path s^*, having links o and d as origin and destination respectively. Similar as for deriving the likelihoods of sensor observation paths, we start with calculation of $q(i|o, d, \emptyset)$; the expected number of link arrivals given that an individual has not been observed by a single sensor. We can show that

$$q(i|o,d,\emptyset) = (\delta_{i,o} + q^0(i|o,d)\cdot(1-\theta_i))\cdot\frac{P(\emptyset|i,d)}{P(\emptyset|o,d)}, \tag{17}$$

where $\delta_{i,o}$ is the Kronecker delta, which equals 1 if $i = o$ and 0 if $i \neq o$. The terms $P(\emptyset|i, d)$ and $P(\emptyset|o, d)$ are the empty sensor observation path likelihoods starting from links i and o respectively (see Section 6.1). The derivation of this formula can be found in Appendix B. To find the expected link utilization in case of a non-empty sensor observation path s^*, we first define

$$\hat{q}(i, s^*|o, d) = q(i|o, d, s^*) \cdot P(s^*|o, d) \tag{18}$$

Similar as in Section 6.2, $\hat{q}(i, s^*|o, d)$ can be expressed recursively:

$$\hat{q}(i, (s_1^*, s_2^*, \ldots, s_n^*)|o, d) = K_1 + K_2, \tag{19}$$

with

$$K_1 = (\delta_{i,o} + q^0(i|o, d) \cdot (1 - \theta_i)) \cdot P((s_1^*, s_2^*, \ldots, s_n^*)|i, d) \tag{20}$$

$$K_2 = \sum_{l \in \mathcal{L}_S(s_1^*)} \left(q^0(l|o, d) \cdot \theta_l \cdot \hat{q}(i, (s_2^*, \ldots, s_n^*)|l, d) \right). \tag{21}$$

Instead of a complete derivation, we will explain the intuition behind the recursive scheme. Let us assume that the length of our sensor observation path s^* equals ζ. In this case, link i can be visited during $\zeta + 1$ different periods: before the first observation, between the first and second observation, between the second and third observation, and so on, till the period after the last observation. The total expected number of visits of link i will be the sum of the expected number of visits of link i during these $\zeta + 1$ periods. In this light, the term K_1 counts the expected number of visits before the first observation from the remaining sensor observation path (see the analogy with (17)). The term K_2 recursively adds the expected number of visits of link i that occur after the first observation of the remaining sensor observation path (see the analogy with (13)). Finally, $q(i|o, d, s^*)$ can be easily computed from $\hat{q}(i, s^*|o, d)$ using Eq. (18). Cumulative link flows can be estimated by summing the link utilization for each individual sensor observation path. These link flows do not represent absolute values but have to be interpreted in a relative way, since not every trip is necessarily being recorded. This relative interpretation can already provide valuable insights into, for instance, the relative popularity of different routes connecting the same origin and destination. To estimate absolute cumulative link flows, the method has to correct for the amount of non-recorded trips. It depends on the application and the availability of other data sources (such as counting sensors for specific cross-sections), whether a simple correction can be applied. One example could be a correction factor that is the inverse of the fraction of festival visitors that downloaded the festival app.

At this point it is worth to mention that another technique exists that aims to reconstruct a route from sensor observations. The technique is based on Hidden Markov Models (HMM) and uses the Viterbi Algorithm to find the most likely path to reproduce a sequence of sensor observations (Musa and Eriksson, 2012). One of the major differences is that the HMM-Viterbi method considers the discretized time of the sensor observations as well and herewith indirectly assumes speed distributions of individuals. For pedestrians in an urban or event context, the effect of such implicit speed assumptions on the accuracy of the outcomes is still unclear. Besides this, the HMM-Viterbi method produces a single route as being the most likely one. Our proposed method is a probabilistic one, assigning a utilization value to each link in the network, which makes the method more suitable for aggregation purposes we believe, especially in cases with large gaps. An advantage of the HMM-Viterbi method is the ability to deal with multiple concurrent sensor observations.

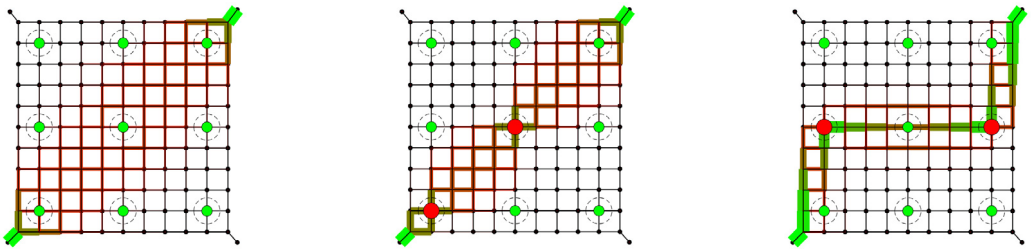
9. Simulated use-case: Link utilization estimation for a single individual

To test the link utilization estimation method, as described in Section 8, we imagine an individual that moves from an origin to a destination in an artificial network, as indicated in Fig. 6. The individuals route choice behaviour is modeled by the next-link probability matrix $\mathbf{P}_d$, which is constructed assuming a utility function that is based on link distances and a penalty for u-turns (see Section 3.2). We defined three imaginary sensor observation paths that can result from the trip. The big red circles in Fig. 6 indicate per scenario the sensor locations where the individual has been observed. The green circles represent sensor locations where the individual has not been observed. The observed link sets were defined according to the *single-node* construction (see Fig. 2(b)). The utilization per link has been estimated using Eq. (18) and the recursive formula (19). The results are shown in Fig. 6, where greener and wider lines indicate higher probabilities that an individual with the given sensor observation path passes this link.

Fig. 6 shows that the calculated flows 'avoid' the sensor locations where the individual has not been observed. This clearly demonstrates the benefit of this method over route reconstruction techniques where only the locations are taken into account where the individual has actually been observed.

To verify the link utilization calculation, we simulated a total of n trajectories from the left-bottom origin to the top-right destination, which we randomly transformed into sensor observation paths, using the detection rates of the sensors (fixed at 0.7). From all randomly generated sensor observation paths, we selected only those that matched the scenario of Fig. 6(b) ((bottom-left, center-center)). The average link utilization over this set of paths gives us an approximation of a person's expected link utilization, given that the person was observed by the (bottom-left) and the (center-center) sensor.

Next, we wanted to know whether this (simulated) true link utilization could be correctly estimated by our method. For this purpose, we calculated the *RMSE* between the simulated and theoretically derived link utilization for different values of n (the unfiltered number of simulated trajectories). The results are shown in Fig. 7 (notice the logarithmic scales). The figure reveals the typical "inverse square root" relation between sample size and sample error of the mean, which supports the belief that our method is able to correctly derive the expected link utilization.



(a) No sensor observations (b) (bottom-left, center-center) (c) (center-left, center-right)

Fig. 6. For three different sensor observation paths, the link utilization has been plotted on the network. The greener and thicker the line, the more likely it is that an individual passes that link, given the bottom-left origin, top-right destination and the sensor observation path as indicated by the big red circles. The green circles represent sensors where the individual has not been observed. It can be seen that the flows towards green circles are relatively small. Default parameter values: $\mu = 1$, $c_{penalty} = -100$ and $\theta = 0.7$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

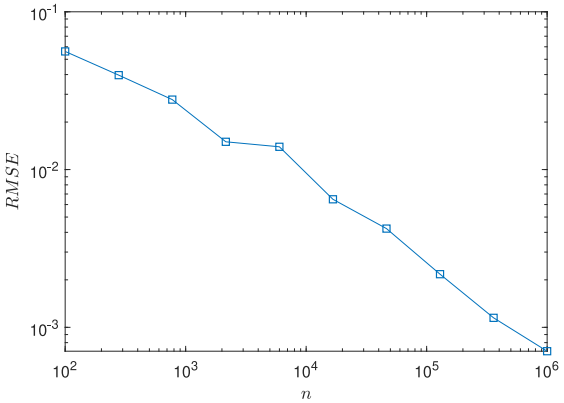


Fig. 7. The *RMSE* between the simulated approximation of link utilization and theoretical link utilization, following the calculations of Section 8. The simulation and theoretical calculation are based on a sensor observation path as in Fig. 6(b). The figure shows a clear decrease of the *RMSE* for an increasing initial sample size n and herewith supports the validity of our link utilization calculation method.

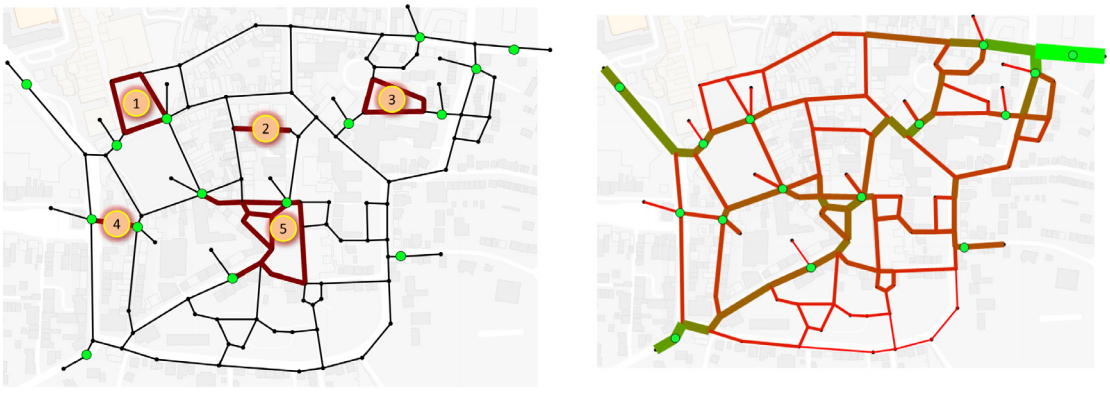
10. Application at a music festival

We tested our route choice model and link utilization estimation method on a data set that was collected during the TT Assen. This Dutch music festival is organized yearly as a side festivity around the Dutch TT motor racing event. In 2018, the festival lasted from June 27, till June 30, and attracted approximately 160,000 visitors. A total of 11 stages were built in the city centre of Assen, where a diversity of musical performances and motor demonstrations were given. In cooperation with the company Connection Systems B.V., we installed Wi-Fi-sensors at 15 different locations in the city centre. Fig. 8(a) shows these sensor locations. Within a radius of 20 m on average, these sensors identify devices in search for a Wi-Fi-network, based on their MAC-address. If the same MAC-address is detected by multiple sensors in the network, we have some insight into the mobility of the person carrying the specific device. The observed link sets of the sensors again were governed by the *single-node* construction (see Fig. 2(b)).

The question that we tried to answer for this specific event was to what extent route choice behavior was influenced by the stage locations. It can be hypothesized that people try to avoid the busy locations when they walk through the city.

10.1. Data cleaning

Wi-Fi-data was collected during the four days of the event. For model estimation, we only used data from the evening of June 28, (starting at 6PM) till the morning of June 29, (ending at 5AM), since stage locations, and herewith link characteristics, differed from day to day, which would complicate our data preparation if we took multiple evenings into account. The raw data tell which MAC-addresses have been observed by which sensors at what times. The data is composed of observations of stationary behavior and observations of travelling behavior. For estimation of the model, we needed the travelling observations, together with the trip origins and destinations. To get this trip information, we processed the raw data as follows:



(a) (b)

Fig. 8. a) The network that was used to assess the mobility during TT Assen festival. The green dots represent the sensor locations (placed at nodes in the network). The yellow circles represent the stage locations on Thursday and the thick red lines represent links that were adjacent to a stage area. b) The estimated cumulative link flows. Parameters: $\mu = 50$, $\beta_{LS} = -0.2$, $c_{penalty} = -100$, $\theta = 0.46$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- A person is assumed to be stationary located at a certain sensor-equipped node when he or she is identified by this sensor more than once in a period that lasts for at least 20 min.
- These stationary locations are set as the origin and destination of a trip. The observation points in between are used to define the sensor observation path of a trip.
- Also, a single observation by one of the four sensors that were placed at the city entrance roads (see Fig. 8(a)) defines a trip origin or destination, since people are expected to only pass these nodes while entering or leaving the city center.
- While travelling, people can be observed multiple times by the same sensor. As some Wi-Fi-sensors were placed close to each other, persons were even occasionally identified alternately by two different sensors during a certain part of his or her trip. To account for this, we discard all observations for which the same person was already identified earlier by the same sensor within a trip.
- While travelling from their origin to their destination, people are assumed to take a route whose distance is not too large compared to the shortest distance possible. For this end, for each sensor observation path, we calculated the shortest cycle-free path, connecting the origin with the destination, that passes through all observation points in the given order. Dividing this path length by the length of the shortest path from origin to destination, which does not necessarily go through the observation points, gives us a lower bound for the so-called detour ratio. To exclude erratic trips, we filtered out all with a detour ratio above 2.5. To find the shortest cycle-free path from an origin to a destination that passes through a set of nodes in a given order, a best-first branch and bound algorithm was adopted (A*-algorithm with branch dependent feasibility constraints). Implementation details are omitted since they fall outside the scope of this paper.
- Since we were only interested in walking behavior, we excluded all observations whose average trip speed was below 0.5 m/s or above 2 m/s. We estimated the average speed using again the distance of the shortest cycle-free path through all observation points.

Since the model requires the origin and destination of a trip to be a link (instead of a node), dummy links have been connected with all sensor-equipped nodes, serving as origin and destination links. The distances of these dummy links have been set to 0. After cleaning, we ended up with 296 sensor observation paths, from which 197 unique ones. For each sensor observation path, the link utilization has been estimated using the formulas in Section 8. Cumulative link flows were derived by summing the estimated link utilization for all sensor observation paths. The cumulative link flows are shown in Fig. 8(b). Notice that these link flows do not represent absolute values, since only a portion of the population has been tracked. Hence, only relative conclusions with respect to the flows can be drawn from the figure.

10.2. Likelihood correction

Only sensor observations strictly between the origin and destination node are defined as being part of a sensor observation path. This is a direct implication from our choice to define the final sensor observation to belong to the stationary phase and not to the travelling phase. Hence, by construction, the link that leads to the destination node is never part of the sensor observation path. This leads to a structural underestimation of the likelihood to reproduce the actual observations, where an observation by the sensor at the final destination might apply to the travelling phase as well. To compensate for this, the likelihood as calculated by Eq. (13) has been corrected by dividing by $(1 - \theta_d)$, where θ_d equals the detection rate of the sensor located at the destination node.

Table 2
Estimated parameters β_{LS}, β_{normal}, β_{stage} and θ and evaluation metrics for the observations between June 28, 6PM and June 29, 5AM.

Parameter	*estimate*	*std. err.*	*t-stat*	*p-value*
β_{LS}	−0.45	0.077	−5.81	6.1e-9
β_{normal}	0.25	0.022	11.4	0
β_{stage}	−0.30	0.056	−5.32	1.1e-7
θ	0.46	0.023	20.1	0

10.3. Analysis

With the collected sensor observation paths, we studied the relation between route choice behavior and stage locations. For this end, we first identified all the links that were part of a stage area. These links are indicated in Fig. 8(a) by thick red lines. The following utility function was defined:

$$v(j|i,o,d) = (-1 + \beta_{normal} \cdot \mathbf{1}_{normal}(j) + \beta_{stage} \cdot \mathbf{1}_{stage}(j) + \beta_{LS} \cdot (\tilde{\mathbf{q}}_{o,d})_l) \cdot \Delta(j) - c_{penalty} \cdot \mathbf{1}_U(i,j) \tag{22}$$

In this expression, $\mathbf{1}_{stage}(j)$ is an indicator function that evaluates to 1 in case link j is part of a stage area (thick red line) and 0 otherwise. The function $\mathbf{1}_{normal}(j)$ returns 1 for a non-stage link and 0 for a stage link. Further, β_{LS} represents the link size attribute value (see Section 3.3). The flow vector $\tilde{\mathbf{q}}_{o,d}$, the second component of the path overlap correction term, is calculated according to (6) using the utility function (22) with β_{LS}, β_{normal} and β_{stage} set to 0. Finally, the function $\mathbf{1}_U(i,j)$ evaluates to 1 in case the transition from link i to link j is a u-turn and 0 otherwise.

The sensors were installed in such a way that their intersections could be observed completely, which makes us assume that each sensor has (approximately) the same detection rate θ. The magnitude of θ, however, was unknown. Therefore, we decided θ to be part of the search space in our optimization process. Thus, we maximized the joint log-likelihood by varying β_{normal}, β_{stage}, β_{LS} and θ. The log-likelihood maximization was performed using MATLAB's function *fmincon*, in which the detection rate θ was constrained to the interval [0,1) and β_{LS} was constrained to the interval $[-1, 1]$. The parameters β_{normal} and β_{stage} were constrained to be smaller than 1 (since preferences for cycles might occur otherwise). The scale parameter μ was kept at a constant value of 50. The results of the optimization are shown in Table 2.

When we analyse the estimated parameters, we first of all recognize the negative value of the link size attribute (β_{LS}), which is in accordance with previous studies (e.g., Fosgerau et al. (2013), Zimmermann et al. (2017)). Regarding the hypothesis, we recognize that the preference for links that are part of a stage area (β_{stage}) is significantly lower than for links that are not part of a stage area (β_{normal}). Although other parameters might play a role as well, the result suggests that people actually tried to avoid the crowded areas while consciously walking to their intended destination.

Finally, some words about the goodness of fit. The value of ρ^2 was calculated as explained in Section 7.3. For the reference parameter set $\boldsymbol{\beta_0}$, we selected zero values for the link size and stage-link attributes and the value $\theta_{est} = 0.46$ for the detection rate. The ρ^2 that was found equals 0.074. A plausible reason for this low value is that prediction of sensor observation paths is fundamentally more difficult than the traditional prediction of routes, since prediction of sensor observation paths is involved with an additional source of stochasticity; the sensor detection rate. Although this stochastic component decreases ρ^2, it has to be kept in mind that we are generally not interested in predicting the actual sensor observation paths, so we do not necessarily consider a low value of ρ^2 as a bad thing.

11. The network-free data approach as a path-based alternative

Bierlaire and Frejinger (2008) proposed a path-based method to estimate route choice models with unprocessed, network-free location data. They introduced the concept of a *Domain of Data Relevance*, which corresponds to a physical region in the network to which a specific observation is relevant. A key element in the method is the a so-called measurement equation, which calculates the probability to observe a certain location sequence, given a certain chosen path. The method was designed to be used with location data, like GPS measurements or self-reported trips. The authors successfully applied their network-free data estimation method on a set of self-reported trips in a network consisting of almost 40,000 unidirectional links.

The network-free data estimation approach is similar to our recursive approach in the sense that it estimates a route choice model with incomplete data. It would therefore be interesting to compare both methods. Except for the fact that the network-free data approach involves generation of a choice set, the method can be applied to our static sensor context in a straightforward way, by interpreting sensor observations as location measurements and observed link sets from sensors as the *Domains of Data Relevance*. We followed the methodology as described in Bierlaire and Frejinger (2008), where the measurement equation results into 1 in case the path crosses all "observed" *Domains of Data Relevance* in the correct order and 0 otherwise. This provides us with an alternative estimation method for the route choice model.

Nevertheless, since this implementation does not use the knowledge of the full sensor network, which includes the locations and detection rates of all sensors, we could expect the method to give biased estimations if applied to such a

Table 3
Estimated parameters $\beta_{est,1}$, $\beta_{est,2}$ and $\beta_{est,3}$ using the naive network-free data approach, the smart network-free data approach and our recursive approach respectively, for different sensor configurations and different values for the true scale parameter β_{true}. Models were estimated using 10,000 simulated sensor observation sequences, with a sensor detection rate $\theta = 0.7$.

Sensor configuration	β_{true}	$\beta_{est,1}$ *(naive network-free)*	$\beta_{est,2}$ *(smart network-free)*	$\beta_{est,3}$ *(recursive)*
Fig. 3(a)	−0.05	−0.127	−0.048	−0.048
Fig. 3(a)	−0.03	−0.001	−0.029	−0.029
Fig. 3(b)	−0.07	−0.028	−0.070	−0.070
Fig. 3(b)	−0.04	−0.017	−0.040	−0.040

static sensor data set. To test this, we simulated 10,000 non-cyclic trajectories in the small network shown in Fig. 3. We chose for a network of such a small size, so that the choice set was unambiguously defined by all possible acyclic paths connecting origin (most left link) and destination (most right link). Trajectories were simulated by repeatedly selecting one out of a of the five possible acyclic paths connecting the origin and the destination. Selection probabilities were determined according to a utility function $v(p) = -\beta \cdot \Delta(p)$, with p a path, β a scaling parameter and $\Delta(p)$ the length of the path. From the trajectories, sensor observation sequences were generated, using a sensor detection rate $\theta = 0.7$. From these sensor observation sequences, we tried to rediscover the utility parameter β by maximum likelihood estimation, using both the naive network-free data approach and our recursive approach, for different scenarios.

The results are shown in Table 3. It shows that the naive smart network-free data approach fails in rediscovering β, which was in line with our expectation. We can improve the estimation performance if we would use a different measurement equation, in which we incorporate knowledge of all sensor locations (also the ones that did not observe the crossing individual) and their detection rates. In fact, if we would use a measurement equation that calculates, for a given path through the network, the exact probability of observing a specific sensor observation sequence, taking all sensors on the path and their detection rates into account, the network-free data approach effectively becomes a path-based version of our recursive approach. To test this, we also implemented the network-free data approach with this so-called smart measurement equation. The same estimations were performed and the results are shown in Table 3. As expected, we see that the smart network-free data approach implementation matches the results of our recursive approach, which shows that the network-free data approach with smart use of the full sensor network information, indeed turns into a path-based version of our recursive approach.

The benefit of using the recursive approach over the smart network-free data approach is that the recursive approach does not require generation of a path choice set. Especially in case of sparse sensor observations, it could be a challenge to generate plausible alternatives. On the other hand, the smart network-free data approach is expected to be computationally more efficient, especially for very large networks.

12. Computational complexity

In this section, we evaluate the computational complexity of the likelihood calculation. In order to calculate the likelihood of a sensor observation sequence, formula (13) has to be executed recursively for each (unique) sequence of sensor observations. Although we already claimed that the number of function evaluations does not grow exponentially in the number of sensor observations or network size (see Section 6), computational time is a critical factor and asks for a thorough assessment. For this purpose, we investigated the relation between the computational time and the following three factors: network size, number of sensors and number of trips. Networks of different sizes and with different numbers of sensors (with observed link sets defined according to the *single-node* construction) were generated according to the evolution scheme as depicted by Fig. 10.

For different scenarios, we measured the processing time to calculate the likelihood to reproduce observation paths that were simulated by adopting (14), with β_{R1} and β_{R2} equal to 0, as the utility function and assuming a detection rate $\theta = 0.8$ for all sensors, a scaling parameter $\mu = 1$ and a u-turn penalty $c_{penalty} = -100$. Although the full estimation machinery has been implemented in MATLAB, we re-implemented the likelihood calculation in Python. The major reason for this was that the recursive scheme could not be efficiently implemented in MATLAB. Although the performance of the MATLAB implementation was sufficient for the use-case (Section 10), we considered it worthless for a critical assessment of the computational efficiency.

Results of the computational tests are shown in Fig. 9. The reported computational times are separated into three components:

- Pre-processing time **P**: the time to calculate next-link probabilities $\mathbf{P}_d$, given a utility function (Eqs. (3) till (5)). These equations are part of the original Recursive Logit model formulation. For large networks, the calculation time is dominated by solving the linear system (4). This *pre-processing time* needs to be spent only once per unique destination in the set of observed trips.

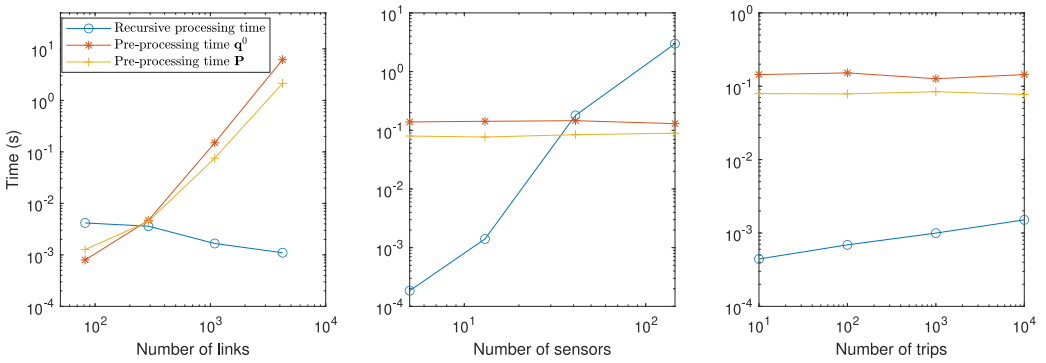


Fig. 9. Graphs showing the relation between the times to calculate the log-likelihood and the following three factors: network size (number of links), number of sensors and number of observed trips. The scenario that we used as the base-case consists of 13 sensors, 1090 links and 10,000 trips. Other parameters: $\theta = 0.8$, $\mu = 1$, $c_{penalty} = -100$.

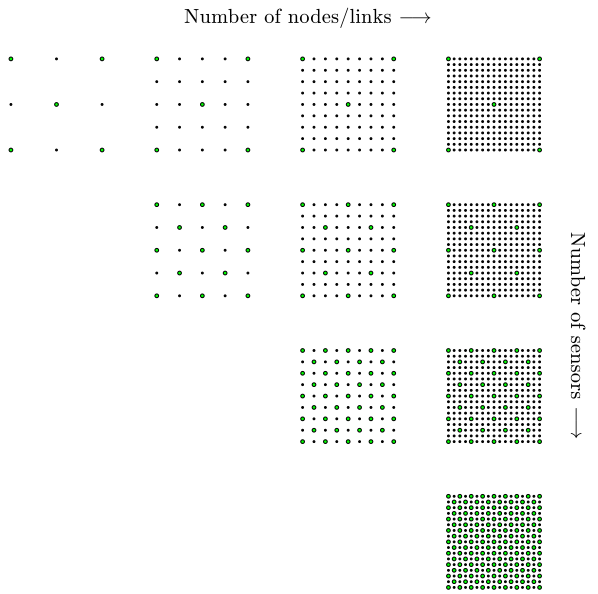


Fig. 10. The scheme that is used to generate networks of different sizes and with different numbers of sensors. All dots represent a network node. Large green dots represent a sensor-equipped node. Horizontally and vertically adjacent pairs of nodes are connected by two directed links (one per direction), but for clarity, these links are not shown in the figure. The demand is specified as a number of people that travel from the top-left node to the bottom-right node. To model the demand, a source and sink dummy link with zero length are connected with the top-left and bottom-right node respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Pre-processing time $\mathbf{q}^0$: the time to calculate the unobserved link flows, given the next-link probabilities $\mathbf{P}_d$ and the link detection rates (Eqs. (A.2) till (A.5)). For large networks, the calculation time is dominated by the matrix inversion in Eq. (A.5). This pre-processing time needs to be spent only once per unique destination in the set of observed trips.
- Recursive processing time: the time needed to run the recursive part of the likelihood calculation (Eq. (13)).

Fig. 9 shows that the combined *pre-processing time* does not depend on the number of sensors or the number of trips. This is in line with our expectations, since the amount and complexity of these calculations do not depend on these factors. The *pre-processing time* does however depend on the network size. For a network with 4226 links, the *pre-processing time* on a single-core running Python kernel was 8.3 s, which is quite a substantial duration. On the other hand, if we compare the two different components of the *pre-processing time*, we conclude that the original RL-model related calculation time (*pre-processing time* $\mathbf{P}$) and the additional *pre-processing time* $\mathbf{q}^0$ are in the same order of magnitude.

The *recursive processing time* depends on all three factors. For an increasing network size (and a fixed number of sensors), the sensor observation paths are expected to become shorter because it gets less likely for a person to cross a sensor-equipped node. As a result, we see that the *recursive processing time* goes down for an increasing network size. For an increasing number of trips we see that the *recursive processing time* increases as well. Although one could expect a proportional increase, the relation is actually less than proportional. This has to do with the fact that we cache and re-use

calculated likelihoods, as explained in Section 6. The more trips we generate, the more we will benefit from this so-called memoization, resulting in a less than proportional relation. Finally, the number of sensors has a clear effect on the computation time. For a total of 145 sensors (and 1090 links and 10,000 trips), the *recursive processing time* equals 3.0 s. Fig. 9 (notice the log-log scale) supports our claim that the relation between the number of sensors and the *recursive processing time* is less than exponential, but still the effect of the number of sensors is substantial.

To summarize the results of the computational tests, we state that the method is likely to be applicable up to networks that contain up to a few thousand links and a few hundred sensors, depending on the number of observations and the complexity of the utility function.

13. Discussion on applications

This section discusses our estimation approach from an application perspective. The section starts with an assessment of the two model requirements, as introduced in Section 3.1, in terms of their implications on the applicability of the model. The section concludes with some considerations regarding how the method could be adapted to allow for a fused data approach, in which static sensor data is enriched with location data.

The first requirement that has been posed for the model formulation is that each link can be observed by at most one sensor (Section 3.1). This requirement has an important implication for the application context. Strictly, it means that sensor detection areas should not overlap, which appears to be not always feasible, especially in the case of Wi-Fi sensors. In the Mysteryland use-case, we also suffered from this problem, where at some points in the network people were tracked by two sensors simultaneously, although the sensors were placed quite far apart from each other. This problem can be solved pragmatically by assigning a person that is observed by multiple sensors to the one with the strongest signal strength. Another solution direction might be to introduce dummy sensors that account for overlapping areas. For a limited amount of overlap of sensor detection areas, these workarounds are supposed to be sufficiently effective. However, for networks that are densely equipped with sensors, causing a substantial overlap of detection areas, the recursive method is supposed to be inappropriate, both for estimating model parameters and for estimating cumulative link flows. In such cases, Hidden Markov Model approaches, like the earlier mentioned method of Musa and Eriksson (2012), could be more promising. The other way around, Hidden Markov Model approaches are supposed to perform worse when sensor observations become more sparse, since these models do not deal well with situations where a person is not being observed at all.

The second requirement that we posed on the model formulation deals with the fact that every possible acyclic path through a sensor detection area should have exactly one link in the constructed sensor observed link set (Section 3.1). As mentioned before, our recursive method effectively calculates likelihoods of sensor crossings, instead of likelihoods of exact sensor observations. This implies that we cannot benefit from the information that is carried by single sensor observations, like the signal strength, or duration. To some extent, we can configure path-specific detection rates, especially when we use the *dummy-link* construction. Nevertheless, we have to conclude that our method is not designed for those applications where the interactions between signal strengths and infrastructure carry crucial information.

Finally, we would like to discuss the possibility to combine the data from static sensors with for instance GPS location data. Our recursive method is designed to deal with static sensors with arbitrary detection rates. Examples of these sensors include Wi-Fi sensors and Bluetooth sensors. In many real-life applications, static sensor observations are the only data available. In some cases, however, the static sensor data can be enriched with location data. An example could be an event where visitors download a smartphone app which collects both location data and beacon encounters. With some minor modifications, the recursive method should be able to facilitate this fused data approach. In brief, combined likelihoods of location observations and static sensor observations could be calculated by multiplying the static sensor observation likelihoods with the already calculated unobserved flows on the links where the location measurements apply to. More research is needed to get a better understanding of the potential of fusing location data and static sensor data, with respect to estimating cumulative link flows and route choice model parameters.

14. Conclusions and future steps

This paper has presented a method to estimate a link-based discrete route choice model with observations from static sensors. More generally, the methodology can be applied to any choice problem that can be represented as a graph and for which only a subset of the possible states can be observed. An expression for the likelihood to reproduce observed sensor paths has been derived and this expression is used in a log-likelihood maximization, in order to find the optimal parameter values. The estimation method has been successfully applied on a data set of simulated trajectories through a small network, as well as on a collection of Wi-Fi-traces that were collected during a Dutch music festival.

Besides the method to estimate a route choice model, we described a probabilistic link utilization estimation method. This method provides us the expected utilization of each link in the network, given the sequence of sensors by which an individual was observed. A key point of the method, with respect to shortest path approaches, is that it exploits all spatial knowledge that we have regarding the sensor locations, also in case an individual was not observed by a particular sensor. The method correctly prescribes the decreased probability that an individual passes a sensor location where he was not observed.

One of the major challenges, which we might address in future research, is to develop a method to select good sensor positions. For route choice model estimation, a specific sensor configuration can either be suitable or unsuitable, depending on link characteristics and the parameters to be estimated. A possible approach to assess a sensor configuration is to analyse the amount of collinearity of the derivatives of the vector with all observation path likelihoods with respect to the different parameters. A high degree of collinearity would imply a lower accuracy of certain parameter predictions. Also for link flow estimation, selection of the sensor positions plays a crucial role. In practice, one might choose to place a sensor at locations where the most accurate estimates are desired. Flow estimates of the links that are not observed will most likely have a lower accuracy.

Declaration of Competing Interest

No potential conflict of interest was reported by the authors.

CRediT authorship contribution statement

Tim P. van Oijen: Conceptualization, Investigation, Data curation, Formal analysis, Software, Writing - original draft, Writing - review & editing. **Winnie Daamen:** Conceptualization, Investigation, Data curation, Writing - review & editing. **Serge P. Hoogendoorn:** Conceptualization, Investigation, Writing - review & editing.

Acknowledgement

The research presented in this paper is part of the research program "Allegro: Unravelling slow mode travelling and traffic: with innovative data to a new transportation and traffic theory for pedestrians and bicycles" (ERC Grant Agreement no. 669792), a Horizon 2020 project which is funded by the European Research Council. The authors would like to thank Dorine Duives for her lead role in collecting the data at the TT Festival 2018, the organisers of the TT Festival, in particular Sander ten Bosch, Tamara Boor and Rowan Bruins, for their cooperation and support and Connection Systems, in particular Jouke Wiltenburg and Jos Smits, for the installation and operation of the Wi-Fi sensors.

Appendix A. Calculation of expected number of unobserved link arrivals $q^0(i|o, d)$

The expected number of unobserved link arrivals $q^0(i|o, d)$ is defined as

$$q^0(i|o,d) = \mathbb{E}(k^0(i)|o,d), \tag{A.1}$$

where $k^0(i)$ equals the number of times that an individual arrives unobserved at link i, when moving from link o to link d. The general approach to calculate $q^0(i|o, d)$ is to calculate link flows that result from 'pumping' one unit of flow into the network at the origin link o. Each flow that passes a sensor-equipped link will be lowered according to its detection rate. As a result, the flow through a link represents the expected number of times that an individual will arrive at the specific link when travelling from o to d, while being unobserved so far.

To formalize this idea, we introduce a matrix $\mathbf{P}^*_d$, whose entries are defined as follows:

$$(\mathbf{P}^*_d)_{i,j} = (1-\theta_i)\, p_{i,j,d} \tag{A.2}$$

The rationale behind the formulation of matrix $\mathbf{P}^*_d$ is that each individual that reaches a link i in the observed link set $\mathcal{L}^*$ has a chance of $(1-\theta_i)$ of not being detected by the particular sensor. Only in that case the individual is still a candidate for travelling unobserved to the destination. Next, we define the vector $\mathbf{g}_o$ as the productions per link. As explained before, it is defined to have the value 1 in the origin link o and 0 elsewhere. Finally, we define the vector $\mathbf{q}^0_{o,d}$, containing the link flows to be calculated. Conservation of mass now tells us that

$$\mathbf{q}^0_{o,d} = (\mathbf{P}^*_d)^T \mathbf{q}^0_{o,d} + (\mathbf{P}_d)^T \mathbf{g}_o. \tag{A.3}$$

In this equation, the production vector $\mathbf{g}_o$ is multiplied with the transposed next-link probability matrix $\mathbf{P}_d$. This has to deal with the fact that we define the start of the trip directly after leaving the origin link. Therefore, instead of pumping a single unit of flow into the origin link, we effectively pump flows into its neighbour links, according to the next-link probabilities $\mathbf{P}_d$.

Rearranging shows that the flow $\mathbf{q}^0_{o,d}$ can be calculated by solving the linear system of equations

$$\left(\mathbf{I} - (\mathbf{P}^*_d)^T\right)\mathbf{q}^0_{o,d} = (\mathbf{P}_d)^T \mathbf{g}_o, \tag{A.4}$$

which gives

$$\mathbf{q}^0_{o,d} = \left(\mathbf{I} - (\mathbf{P}^*_d)^T\right)^{-1} (\mathbf{P}_d)^T \mathbf{g}_o. \tag{A.5}$$

The expected number of unobserved link arrivals, $q^0(i|o, d)$, now simply equals the ith component of the flow vector $\mathbf{q}^0_{o,d}$:

$$q^0(i|o,d) = (\mathbf{q}^0_{o,d})_i \tag{A.6}$$

Appendix B. Derivation of Eq. (17)

Let us take an arbitrary route of an individual from origin o to destination d. We define s^* as the observed sensor path, $k(i)$ as the number of times that the individual travelled on link i and $k^0(i)$ as the number of times that the individual arrived at link i while being unobserved so far. We want to prove that

$$q(i|o,d,\emptyset) = (\delta_{i,o} + q^0(i|o,d)\cdot(1-\theta_i))\cdot\frac{P(\emptyset|i,d)}{P(\emptyset|o,d)}, \tag{B.1}$$

where $q(i|o, d, \emptyset)$ is the expected number of times that link i is visited, having links o and d as origin and destination respectively, given the sensor observation path s^*. We start our derivation with the definition of the conditional expected value $q(i|o, d, \emptyset)$:

$$q(i|o,d,\emptyset) = \frac{1}{P(\emptyset|o,d)}\cdot\sum_{x=1}^{\infty}\left\{x\cdot P((k(i)=x)\cap(s^*=\emptyset)|o,d)\right\} \tag{B.2}$$

Now, let us first assume that $i \neq o$. In this case, $k(i) = k^0(i)$ for all possible routes for which $s^* = \emptyset$, so we may write

$$q(i|o,d,\emptyset) = \frac{1}{P(\emptyset|o,d)}\cdot\sum_{x=1}^{\infty}\left\{x\cdot P\Big((k^0(i)=x)\cap(s^*=\emptyset)|o,d\Big)\right\} \tag{B.3}$$

Each time an individual arrives at the start of link i, while not being observed so far, the probability of arriving unobserved at the destination d equals $(1-\theta_i)$ (non-detection rate at link i) times $P(\emptyset|i, d)$ (probability to travel unobserved to the destination, starting from i). Hence,

$$q(i|o,d,\emptyset) = \frac{1}{P(\emptyset|o,d)}\cdot\sum_{x=1}^{\infty}\left\{x\cdot P(k^0(i)=x|o,d)\cdot(1-\theta_i)\cdot P(\emptyset|i,d)\right\} \tag{B.4}$$

$$= \frac{1}{P(\emptyset|o,d)}\cdot\sum_{x=1}^{\infty}\left\{x\cdot P(k^0(i)=x|o,d)\right\}\cdot(1-\theta_i)\cdot P(\emptyset|i,d) \tag{B.5}$$

We notice that

$$\sum_{x=1}^{\infty}\left\{x\cdot P(k^0(i)=x|o,d)\right\} = \mathbb{E}(k^0(i)|o,d) = q^0(i|o,d), \tag{B.6}$$

which finally gives us:

$$q(i|o,d,\emptyset) = q^0(i|o,d)\cdot(1-\theta_i)\cdot\frac{P(\emptyset|i,d)}{P(\emptyset|o,d)} \tag{B.7}$$

For $i \neq o$, Eq. (B.1) has been shown to be correct. If we set $i = o$, we notice that $k(i) = k^0(i) + 1$ for all possible routes for which $s^* = \emptyset$, since the individual starts his route already at link i. So, we get

Following the same steps as for $i \neq o$, we finally come to

$$q(i|o,d,\emptyset) = \Big(1 + q^0(i|o,d)\cdot(1-\theta_i)\Big)\cdot\frac{P(\emptyset|i,d)}{P(\emptyset|o,d)}, \tag{B.8}$$

which completes the proof.

References

Abdel-Aty, M.A., Kitamura, R., Jovanis, P.P., Reddy, P., Vaughn, K.M., 1995. New approach to route choice data collection : multiphase, computer-aided telephone interview panel surveys using geographic information systems data base. Transp. Res. Rec. (1493) 159–169.

Ben-Akiva, M., Bierlaire, M., 1999. Discrete Choice Methods and their Applications to Short Term Travel Decisions. Springer US, Boston, MA, pp. 5–33.

Ben-Akiva, M.E., Lerman, S.R., 1985. Discrete Choice Analysis Theory and Application to Travel Demand. MIT press, Cambridge, MA.

Bierlaire, M., Frejinger, E., 2008. Route choice modeling with network-free data. Transp. Res. Part C 16 (2), 187–198. doi:10.1016/J.TRC.2007.07.007.

Broach, J., Dill, J., Gliebe, J., 2012. Where do cyclists ride? A route choice model developed with revealed preference GPS data. Transp. Res. Part A 46 (10), 1730–1740. doi:10.1016/j.tra.2012.07.005. arXiv: 1011.1669v3.

Cascetta, E., Nuzzolo, A., Russo, F., Vitetta, A., 1996. A modified logit route choice model overcoming path overlapping problems: specification and some calibration results for interurban networks. In: Lesort, J.B. (Ed.), Transportation and Traffic Theory, Proceedings from the Thirteenth International Symposium on Transportation and Traffic Theory, Lyon, France, Pergamon, pp. 697–711.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to Algorithms. MIT press, Cambridge, MA.

Danalet, A., 2015. Activity Choice Modeling for Pedestrian Facilities, vol. 6806. EPFL, Lausanne doi:10.5075/epfl-thesis-6806.

Danalet, A., Farooq, B., Bierlaire, M., 2014. A bayesian approach to detect pedestrian destination-sequences from WiFi signatures. Transp. Res. Part C 44, 146–170. doi:10.1016/J.TRC.2014.03.015.

Fosgerau, M., Frejinger, E., Karlstrom, A., 2013. A link based network route choice model with unrestricted choice set. Transp. Res. Part B 56, 70–80. doi:10.1016/J.TRB.2013.07.012.

Galama, I.M., 2015. Route Choice Behaviour at Mass Events. Delft University of Technology Master thesis. uuid:5f17df87-14f3-4154-a4d4-6bfe331397fa

van den Heuvel, J., Voskamp, A., Daamen, W., Hoogendoorn, S.P., 2015. Using bluetooth to estimate the impact of congestion on pedestrian route choice at train stations. In: Chraibi, M., Boltes, M., Schadschneider, A., Seyfried, A. (Eds.), Traffic and Granular Flow '13. Springer International Publishing, Cham, pp. 73–82.

Huang, Z., Huang, Z., Zheng, P., Xu, W., 2018. Calibration of C-logit-based SUE route choice model using mobile phone data. Information 9 (5). doi:10.3390/info9050115.
Iqbal, M.S., Choudhury, C.F., Wang, P., González, M.C., 2014. Development of origin-destination matrices using mobile phone call data. Transp. Res. Part C 40, 63–74. doi:10.1016/j.trc.2014.01.002.
Kurkcu, A., Ozbay, K., 2017. Estimating pedestrian densities, wait times, and flows with Wi-Fi and bluetooth sensors. Transp. Res. Rec. 2644, 77–82.
Larsen, J.E., Sapiezynski, P., Stopczynski, A., Mørup, M., Theodorsen, R., 2013. Crowds, bluetooth, and rock'n'roll: understanding music festival participant behavior. In: Proceedings of the 1st ACM International Workshop on Personal Data Meets Distributed Multimedia. ACM, New York, NY, USA, pp. 11–18. doi:10.1145/2509352.2509399.
Leontiadis, I., Lima, A., Kwak, H., Stanojevic, R., Wetherall, D., Papagiannaki, K., 2014. From cells to streets. In: Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies - CoNEXT '14. ACM, Sydney, Australia, pp. 121–132. doi:10.1145/2674005.2674982.
Lesani, A., Miranda-moreno, L., 2018. Development and testing of a real-time WiFi-bluetooth system for pedestrian network monitoring, classification, and data extrapolation. IEEE Trans. Intell. Transp. Syst. PP, 1–13. doi:10.1109/TITS.2018.2854895.
Lu, W., Scott, D.M., Dalumpines, R., 2018. Understanding bike share cyclist route choice using GPS data: comparing dominant routes and shortest paths. J. Transp. Geogr. 71 (July), 172–181. doi:10.1016/j.jtrangeo.2018.07.012.
Mahmassani, H., Peeta, S., 1993. Network performance under system optimal and user optimal assignments: implications for advanced traveler information systems. Transp. Res. Rec. 1408 (August 2015), 183–193.
Mai, T., Bastin, F., Frejinger, E., 2018. A decomposition method for estimating recursive logit based route choice models. EURO J. Transp. Logist. 7 (3), 253–275. doi:10.1007/s13676-016-0102-3.
Mai, T., Fosgerau, M., Frejinger, E., 2015. A nested recursive logit model for route choice analysis. Transp. Res. Part B 75, 100–112. doi:10.1016/j.trb.2015.03.015.
Menghini, G., Carrasco, N., Schüssler, N., Axhausen, K.W., 2010. Route choice of cyclists in Zurich. Transp. Res. Part A 44 (9), 754–765. doi:10.1016/j.tra.2010.07.008.
Musa, A.B.M., Eriksson, J., 2012. Tracking unmodified smartphones using Wi-Fi monitors. In: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems. ACM, New York, NY, USA, pp. 281–294.
Oyama, Y., Hato, E., 2018. Link-based measurement model to estimate route choice parameters in urban pedestrian networks. Transp. Res. Part C 93, 62–78. doi:10.1016/J.TRC.2018.05.013.
Ramming, M.S., 2001. Network Knowledge and Route Choice. Massachusetts Institute of Technology Ph.D. thesis.
Ton, D., Duives, D., Cats, O., Hoogendoorn, S., 2018. Evaluating a data-driven approach for choice set identification using GPS bicycle route choice data from Amsterdam. Travel Behav. Soc. 13 (July), 105–117. doi:10.1016/j.tbs.2018.07.001.
Versichele, M., Neutens, T., Delafontaine, M., Van de Weghe, N., 2012. The use of bluetooth for analysing spatiotemporal dynamics of human movement at mass events: a case study of the ghent festivities. Appl. Geogr. 32 (2), 208–220. doi:10.1016/j.apgeog.2011.05.011.
Wang, Y., Correia, G.H.d.A., van Arem, B., Timmermans, H.J., 2018. Understanding travellers' preferences for different types of trip destination based on mobile internet usage data. Transp. Res. Part C 90 (March), 247–259. doi:10.1016/j.trc.2018.03.009.
Yoon, J., Noble, B.D., Liu, M., Kim, M., 2006. Building realistic mobility models from coarse-grained traces. In: Proceedings of the 4th International Conference on Mobile Systems, Applications and Services. ACM, New York, NY, USA, pp. 177–190. doi:10.1145/1134680.1134699.
Zimmermann, M., Mai, T., Frejinger, E., 2017. Bike route choice modeling using GPS data without choice sets of paths. Transp. Res. Part C 75, 183–196. doi:10.1016/J.TRC.2016.12.009.