PLOS ONE

RESEARCH ARTICLE

Large-scale simulation of traffic flow using Markov model

Renátó Besenczi*, Norbert Bátfai†, Péter Jeszenszky, Roland Major, Fanny Monori, Márton Ispány

Department of Information Technology, University of Debrecen, Debrecen, Hungary

† Deceased.
* besenczi.renato@inf.unideb.hu

OPEN ACCESS

Citation: Besenczi R, Bátfai N, Jeszenszky P, Major R, Monori F, Ispány M (2021) Large-scale simulation of traffic flow using Markov model. PLoS ONE 16(2): e0246062. https://doi.org/10.1371/journal.pone.0246062

Editor: Yanyong Guo, Southeast University, CHINA

Received: September 1, 2020

Accepted: January 12, 2021

Published: February 9, 2021

Data Availability Statement: The data underlying the results presented in the study are available from https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i and https://www.openstreetmap.org/.

Funding: The publication is supported by the EFOP-3.6.1-16-2016-00022 project. The project is co-financed by the European Union and the European Social Fund. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Modeling and simulating movement of vehicles in established transportation infrastructures, especially in large urban road networks is an important task. It helps in understanding and handling traffic problems, optimizing traffic regulations and adapting the traffic management in real time for unexpected disaster events. A mathematically rigorous stochastic model that can be used for traffic analysis was proposed earlier by other researchers which is based on an interplay between graph and Markov chain theories. This model provides a transition probability matrix which describes the traffic's dynamic with its unique stationary distribution of the vehicles on the road network. In this paper, a new parametrization is presented for this model by introducing the concept of two-dimensional stationary distribution which can handle the traffic's dynamic together with the vehicles' distribution. In addition, the weighted least squares estimation method is applied for estimating this new parameter matrix using trajectory data. In a case study, we apply our method on the Taxi Trajectory Prediction dataset and road network data from the OpenStreetMap project, both available publicly. To test our approach, we have implemented the proposed model in software. We have run simulations in medium and large scales and both the model and estimation procedure, based on artificial and real datasets, have been proved satisfactory and superior to the frequency based maximum likelihood method. In a real application, we have unfolded a stationary distribution on the map graph of Porto, based on the dataset. The approach described here combines techniques which, when used together to analyze traffic on large road networks, has not previously been reported.

Introduction

In the past decade, research and development of smart city applications have become an active topic [1, 2]. These services contain solutions such as intelligent city planning, crowdsourcing, as well as crisis and disaster management [3]. These applications will also both generate and make use of big data which will arise from the wide availability of cloud computing and IoT applications [4]. By the year 2050, 68% of Earth's population is expected to live in urban areas [5]. City infrastructures will face new challenges from many factors; one such factor is urban

traffic. Moreover, a solution for the problem of air pollution and congestion is highly demanding [6–8]. In the recent years, the research and development of intelligent transportation systems (ITS) in the context of smart cities have become a vivid topic [9, 10]. In the near future, a smart city ITS application may also have a requirement to support the operation of both self-driving and electric cars [11–13].

This research follows and contributes to our development of a traffic simulation platform initiative called rObOCar World Championship (or OOCWC for short) [13–17]. The OOCWC is a multiagent-oriented environment for creating urban traffic simulations and for investigating the relationship between smart cities and self-driving cars. The traffic simulations are performed by one of its components called *Robocar City Emulator* (RCE). We extract geographical information from OpenStreetMap (OSM) and transform this data into a routing map graph. The simulation takes place on a rectangular part of the OSM. The traffic simulation model of the RCE is based on the Nagel-Schreckenberg cellular automata model [18]. We slice all graph edges for parts 3 meters long, so the length of each cell is $l = 3m$. Each edge has only one lane and up to one car can occupy a cell at every time step $\Delta t = 0.2s$; therefore, each simulation unit moves with fixed speed $v = 54km/h$. During the simulation, we can observe how the distribution of the cars changes. In the original implementation, the simulation algorithm moves the cars by random walk. So, when a car arrives at a graph vertex (i.e. intersection), it selects the next edge (i.e. next road segment) according to uniform distribution and is delivered to the next edge if the first cell of the next edge is free. This model is somewhat similar to that is used in [7]. One statistic that can represent this distribution to test the stationarity is the order of the streets based on the number of cars on them. An important aspect is that the order of the street should remain the same during the simulation when it is already in a steady state. In paper [14], we showed that in the original edition of the OOCWC the order of the streets changes almost randomly even when the simulation has been running for a long time, so the requirement of stationarity does not hold. In this paper, a method is proposed to answer this problem. For a detailed description of the operation of RCE, see paper [13]. There exist several traffic simulation platforms (Multi-Agent Transport Simulation [19], Simulation of Urban Mobility [20], Aimsun (see https://www.aimsun.com/), PTV Vissim (see http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/)). Although these applications are widely used in traffic analysis and planning, the main focus of their simulation algorithms is on microscopic traffic events. In contrast, our software system focuses only on the traffic flow of the whole city, or, to be more precise, the traffic graph.

A fundamental requirement in developing a traffic simulation algorithm which controls the simulated cars is to hold the real distribution of cars in a stationary way, see our previous paper [14]. Another aspect is how we are able to fit this algorithm by estimating its parameters based on real data, which has the form of trajectories. Our result presented in this paper is an answer to these problems. In [21], see also [22] and [23], a stochastic model is proposed which can handle the traffic in an urban network by using a mathematically rigorous method. This model is based on discrete time Markov chain on the road graph which plays the role of the state space. In the traffic interpretation, the transition probability matrix describes the dynamic of the traffic while its unique stationary distribution corresponds to the traffic equilibrium or steady state on the road network. In this steady-state, the distribution of vehicles remains invariant locally in time under the transport dynamic. Thus, this stationary distribution of the Markov chain can be interpreted as the momentary "true" distribution of the vehicles on the road network.

Note that the joint application of Markov chains and large graphs to analyze the behavior of complex systems is well known in several fields, e.g., distributed systems [24], geophysics [25] and biology [26]. Several approaches exist for short-term traffic flow prediction. These models

are based on many techniques including Box-Jenkins time-series analyses with ARIMA model [27–30], Kalman filter theory [31, 32], non-parametric methods (k-NN, kernel, local regression) [33–36], exponential smoothing [37, 38], spectral analysis [39] or wavelets [40–42]. In addition, several approaches use machine learning and data mining techniques, such as support vector regression [43], artificial neural networks [44–46], Bayesian networks [47] or deep learning [48]. Some applications can be found based on computational intelligence techniques, e.g., linear genetic programming [49] or fuzzy logic [50–52], but seldom can we find approaches based on Markov models, see [53] and [21] mentioned previously.

Our contributions in this paper are as follows. Based on [21], we introduce the concepts of a Markov random walk, which describes the motion of an individual vehicle, and Markov traffic, which describes the entire traffic on the road network, respectively. We derive the stationary distribution of the Markov traffic as a multinomial distribution, see formula (3). We present how the ergodic theory of finite Markov chains can be applied to prove the ergodicity of Markov traffic model which implies that complex traffic events can be approximated well by the help of the stationary distribution of a Markov chain on the road network. This result also yields the theoretical ground of our simulation algorithm. We reparametrize the model by introducing the concept of two-dimensional stationary distribution which possesses equi-distributed marginals that are the unique stationary distribution of the transition probability matrix, respectively. To estimate this parameter matrix the weighted least squares (WLS) estimation as a kind of composite (quasi-) likelihood methods is applied, see [54]. In Theorem 2, we show that the WLS estimator of the two-dimensional stationary distribution can be expressed explicitly. Moreover, this estimation method provides a computationally effective technique on a large scale since the MapReduce paradigm can be easily applied to it. Finally, we present how a city-controlled IT solution can be developed which is able to simulate the traffic on a road network that fits to real world data.

Modeling traffic flow by Markov chains on graphs

In this section, we overview a traffic simulation model that uses tools from graph theory and Markov chains. First, we outline the basic concepts in the fields of graph theory and finite Markov chains. Then, we describe the proposed model called "Markov traffic" shortly. Subsection after that is devoted to the ergodicity of Markov traffic model. As a case study, we use a publicly available trajectory dataset, namely, the Taxi Trajectory Prediction dataset (see https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i). After, we outline the key points of how we selected and processed the trajectory data. Finally, we describe how we process OSM data and build a traffic graph from this data.

Road network: Basic concepts and notation

In this subsection, we outline the concepts of graph theory that are necessary for modeling traffic flow. A standard textbook on graph theory is [55].

Let $G = (V, E)$ be a directed graph (digraph) where V and E denote the set of vertices or nodes and the set of directed edges or arcs or arrows of the graph, respectively. In the sequel, vertices are denoted by u, v, w, edges are denoted by e, f, g. For a directed edge $e = (v, w) \in E$ we also use the notation $v \longrightarrow w$. We suppose that G is a simple digraph in the sense that it does not contain multiple arrows and loops. Multiple arrows means two or more edges that connect the same two vertices in the same direction. The edge (v, v), $v \in V$, is called a loop, i.e., it connects the vertex v to itself. The digraph G, called road network in this paper, represents the road system of a city. More precisely, we start from the following definition, see [56].

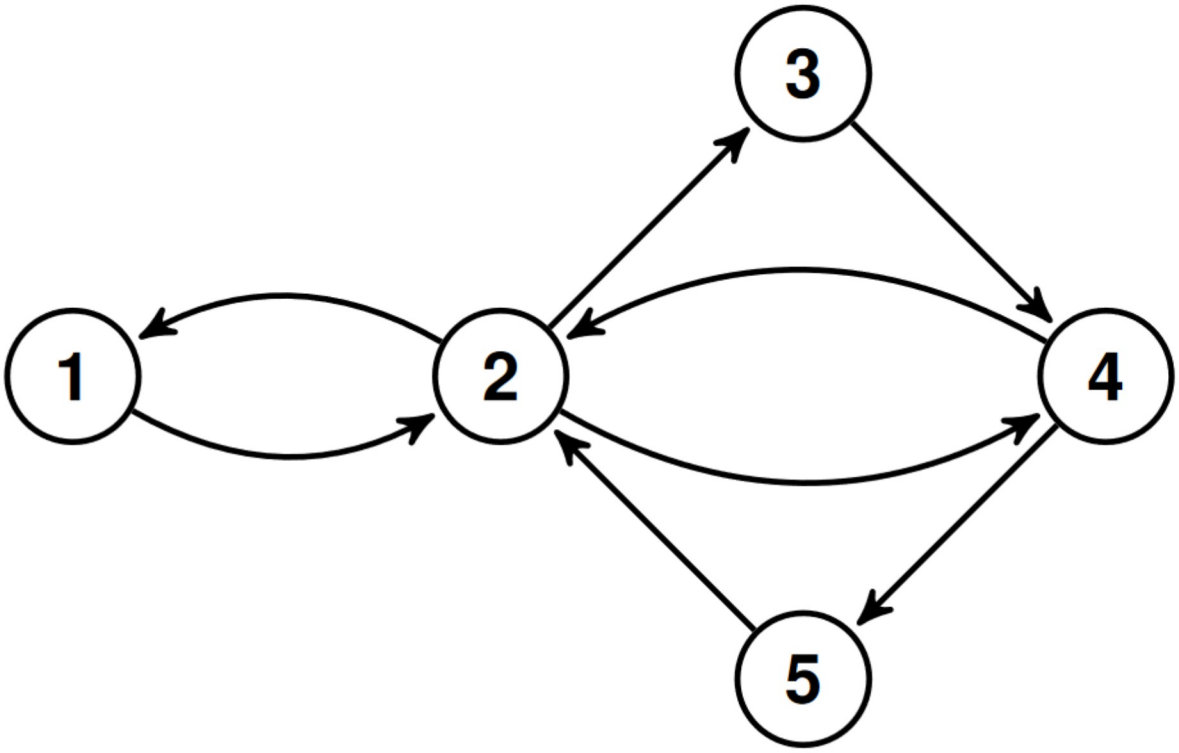


Fig 1. A simple road network.

https://doi.org/10.1371/journal.pone.0246062.g001

Definition 1. A **road network** G is a simple directed graph, $G = (V, E)$, where V is a set of nodes representing the **terminal point**s of road segments, and E is a set of directed edges denoting road segments.

A **road segment** $e = (v, w) \in E$ is a directed edge in the road network graphs, with two terminal points v and w. The vehicle flow on this edge is from v to w.

Note that the simplicity of the graph model of an existing physical road network is clearly guaranteed if the resolution of the network is enough high. The resolution of a road network can be increased by introducing new terminal points on road segments splitting them into smaller road segments. By locking out the loops we can avoid that vehicles can move in an infinite cycle remaining persistently at the same node. Later however, when we define the traffic flow on a road network, we allow "virtual" loops to ensure that vehicles may remain at the same node or edge of the road network after a time step. Let S denote the set of loops in G, i.e., $S := \{(v, v) | v \in V\}$. Since G is simple $E \cap S = \emptyset$. Fig 1 presents a simple example for road network.

For a digraph $G = (V, E)$ another digraph can be associated by the following way. Let the set V' of vertices of this new digraph be the set of directed edges E of G and let the set E' of its directed edges consist of the ordered pair (e, f) where $e, f \in E$ such that there exist $u, v, w \in V$ that $e = (u, v)$ and $f = (v, w)$, i.e., $u \to v \to w$ is a path (dipath) in G of length 2. This associated digraph is called a directed line graph, see Section 4.5 in [55], shortly line digraph, or line road network (network line graph, see [57]), and it is denoted by $L(G) = (V', E')$. The elements of E' can be described by triplets (u, v, w), where $u, v, w \in V$, $(u, v), (v, w) \in E$, and for a directed edge in $L(G)$ we may use the notation $(u, v) \to (v, w)$ too.

The basic difference between the digraph and line digraph views of a road network is that the former assigns the vehicles moving in a city to the vertices while the latter to the edges. One can refer the former as first-order (primal) network while the latter as second-order (dual) network, see [58, 59]. These two kinds of graphs are both useful because in a road network, certain measurements are associated with the crossings (vertices), and certain measurements are associated with the road segments (directed edges). When we are concerned with comparing measurements associated with crossings, then we will be concerned with the adjacency relationships of crossings, and so with the road network. However, when we are concerned with measurements associated with road segments we will be concerned with the adjacency relationships of road segments, and so our analyses will involve the line road network.

The degree distributions of the digraphs G and L(G), respectively, are given in the following way. For $v \in V$ define $v^- := \{e \in E | \exists u \in V: e = (u, v)\}$ and $v^+ := \{e \in E | \exists w \in V: e = (v, w)\}$, i.e., v^- and v^+ are the sets of arrows in and out the node v, respectively. Note that $deg^-(v) = |v^-|$ and $deg^+(v) = |v^+|$ is the indegree and outdegree of v, respectively, where $|\cdot|$ denotes the cardinality of a set. For all $i = 0, 1, 2, \ldots$ define $n_i^+ := |\{v \in V \mid deg^+(v) = i\}|$. Then, the pairs $(i, n_i^+), i = 0, 1, 2, \ldots$, form the frequency histogram for the outdegree distribution of G. The indegree frequency histogram is defined similarly as $(i, n_i^-), i = 0, 1, 2, \ldots$, where $n_i^- := |\{v \in V \mid deg^-(v) = i\}|$. On the other hand, for all $i = 0, 1, 2, \ldots$, define $m_i^+ := \sum_{v \in G_i^+} deg^-(v)$ where $G_i^+ := \{v \in V \mid deg^+(v) = i\}$. Then, the pairs $(i, m_i^+), i = 0, 1, 2, \ldots$, form the frequency histogram for the outdegree distribution of L(G). Note that $n_i^+ = |G_i^+|$ for all i. Similarly, the pairs $(i, m_i^-), i = 0, 1, 2, \ldots$, form the frequency histogram for the indegree distribution of L(G) where $m_i^- := \sum_{v \in G_i^-} deg^+(v)$ and $G_i^- := \{v \in V \mid deg^-(v) = i\}$. (Note that $n_i^- = |G_i^-|$ for all i.) One can easily see that the supports of the two indegree (outdegree) histograms are the same. For the Porto example (described later in this paper), the above mentioned degree distributions can be seen in Fig 2. These histograms corroborate the fact that the Porto's road network, as all city's road network, is a sparse graph since there is no node with higher in- and outdegree than 6 and the ratio of the number of edges and the number of nodes is less than 2, see Fig 5.

Finally, we recall some topological properties of digraph G. For a pair $u, v \in V, u \neq v$, v is reachable from u if there exists a walk $u = v_1 \to v_2 \to \ldots \to v_\ell = v$ where $v_i \in V$ ($i = 1, \ldots, \ell$). A digraph G is said to be strongly connected (diconnected) if every vertex is reachable from every other vertex. Clearly, the line digraph of a strongly connected digraph is also strongly connected. A cycle $C \subset V$ in digraph G is a path $v_1 \to v_2 \to \ldots \to v_\ell \to v_1$ where $v_i \in V, i = 1, \ldots, \ell$, are different nodes. Here $\ell(C) = \ell$ is called the length of C. A digraph G is said to be aperiodic if the greatest common divisor of the lengths of its cycles is one. Formally, the period of G is defined as $per(G) := \gcd\{\ell > 0: \exists C \subset V$ cycle such that $\ell(C) = \ell\}$. Then, G is aperiodic if $per(G) = 1$. One can also see that the line digraph of an aperiodic digraph G is also aperiodic.

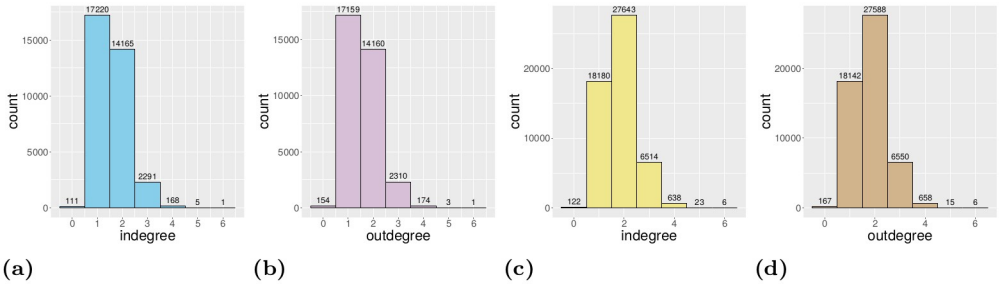


Fig 2. The degree distribution histograms of the Porto map traffic graph. a: Indegree distribution (vertices). b: Outdegree distribution (vertices). c: Indegree distribution (edges). d: Outdegree distribution (edges).

https://doi.org/10.1371/journal.pone.0246062.g002

In a proper traffic, there are vehicles which leave or enter the city. To model these two possibilities V is augmented by a new ideal vertex 0 which denotes the world outside of the city. This approach is similar to that is applied for public transport in [60]. Let $\overline{V} := V \cup \{0\}$. Then, additional directed edges which contains vertex 0 are also added to E. In this case, $(v, 0)$ denotes that the vehicles can leave the city at vertex v, and $(0, v)$ denotes that new vehicles can enter the city at vertex v, where $v \in V$. Let $\overline{E}$ denote the augmentation of E by directed edges for getting into and out of the city. Note that $\overline{E}$ does not contain the loop $(0, 0)$. The augmentation of G is denoted by $\overline{G} = (\overline{V}, \overline{E})$ and it is called the closure of road network G. For $e = (v, w) \in \overline{E}$ we also use the notation $v \to w$. Moreover, each aforementioned concept, e.g., strong connectivity, periodicity, line digraph, given for G can be extended for $\overline{G}$ in a natural way. Note that in the augmented line digraph $\mathrm{L}(\overline{G}) = (\overline{V}', \overline{E}')$ the elements of the edge set $\overline{E}'$ can be described by triplet (u, v, w), where $u, v, w \in \overline{V}$ such that $u \to v \to w$ and if $v = 0$ then $u, w \neq 0$ and if u or w is 0 then $v \neq 0$ because triplets $(0, 0, v)$, $(v, 0, 0)$ and $(0, 0, 0)$ are excluded from $\overline{E}'$. One can easily see that if G is strongly connected (aperiodic) then its closure $\overline{G}$ (as well as $\mathrm{L}(\overline{G})$) is also strongly connected (aperiodic). In the rest of this paper, it is assumed that the road network is closed, i.e., the vehicles can not get into and out of the road system of the city augmented with the ideal vertex 0. Moreover, for the sake of simplicity, only the first-order (primal) network is considered which is denoted by G as well.

We define vectors (functions) and matrices (operators or kernels) on V in the usual way. Let $\boldsymbol{\alpha} : V \to \mathbb{R}$ denote a real function on V and let $\mathscr{F}(V, \mathbb{R})$ denote the set of real functions on V. We also use the notations $\boldsymbol{\alpha}(v) = \boldsymbol{\alpha}_v$ for all $v \in V$ and $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_v)_{v \in V}$. $\mathscr{F} = \mathscr{F}(V, \mathbb{R})$ is a finite dimensional vector space with the usual inner (dot) product. A $T : V \times V \to \mathbb{R}$ real function is called matrix, operator or kernel on V and induces a linear operator on $\mathscr{F}(V, \mathbb{R})$ in the usual way. Moreover, we write $T(\boldsymbol{\alpha}) = T\boldsymbol{\alpha}$ as a matrix-vector product. If the support of T (the set $\{(u, v) | u, v \in V: t_{uv} \neq 0\}$ in $V \times V$) is a subset of E ($E \cup S$) then T is called G-subordinated in strong (weak) sense.

An example for matrix on V is the adjacency matrix $A = (a_{uv})_{u,v \in V}$ of the digraph G where $a_{uv} = 1$ if and only if $(u, v) \in E$ and 0 otherwise. Clearly, the support of A is E, i.e., A is a G-subordinated matrix in strong sense ($a_{vv} = 0$ for all $v \in V$). It is known that G is strongly connected if and only if there is a positive integer k such that the matrix $I + A + \ldots + A^k$ is positive, i.e., all the entries of this matrix are positive. The indegree and outdegree of a vertex v can be expressed by the adjacency matrix as $deg^-(v) = \sum_{u \in V} a_{uv}$ and $deg^+(v) = \sum_{u \in V} a_{vu}$. Introduce the vectors $\boldsymbol{d}^- := (deg^-(v))_{v \in V}$ and $\boldsymbol{d}^+ := (deg^+(v))_{v \in V}$. Then, we have $\boldsymbol{d}^- = A^T \mathbf{1}$ and $\boldsymbol{d}^+ = A\,\mathbf{1}$ where $\mathbf{1} := (1)_{v \in V}$ is the constant unit function. It is well known that the adjacency matrix A of an aperiodic, strongly connected digraph G is primitive, i.e., irreducible and has only one eigenvalue of maximum modulus. Primitivity is equivalent to the following quasi-positivity: there exists $k \in \mathbb{N}$ such that the matrix $A^k > 0$, see Section 8.5 in [61].

Probability distributions and Markov kernels on road networks

In this section, we summarize some basic concepts and results of the theory of finite Markov chains with their interpretations and consequences for traffic flow modeling. Some textbooks on this field are [62] and [63].

A probability distribution (p.d.) on V is the vector $\boldsymbol{\pi} := (\pi_v)_{v \in V}$ where $\boldsymbol{\pi}_v \geq 0$ for all $v \in V$ and $\sum_{v \in V} \pi_v = 1$. That is a p.d. $\boldsymbol{\pi}$ is a normalized $V \to \mathbb{R}_+$ function. We can think of $\boldsymbol{\pi}_v$ as the proportion of the number of vehicles which drive through the crossing v with respect to the whole number of vehicles in the city at a fixed time period. A Markov kernel or transition

probability matrix on V is defined as a real kernel $P \coloneqq (p_{uv})_{u,v\in V}$ such that $p_{uv} \geq 0$ for all $u, v \in V$ and $\Sigma_{v\in V} p_{uv} = 1$ for all $u \in V$, i.e., $\boldsymbol{p}_u \coloneqq (p_{uv})_{v\in V}$ is a p.d. on V for all $u \in V$. The quantity $p_{uv} \in [0, 1]$ is called the transition probability from vertex u to vertex v. The kernel P is said to be G-subordinated if $p_{uv} > 0$ for a pair $u, v \in V$ implies $(u, v) \in E$ or $u = v$, i.e., P as a matrix on V is G-subordinated in the weak sense. It is well known, see [64], that for a Markov kernel P on V, an associated digraph $G_P = (V, E_P)$ can be introduced in the following way: for a pair $u, v \in V$ (where the case $u = v$ is also allowed) $(u, v) \in E_P$ if and only if $p_{uv} > 0$. Thus, P is G-subordinated if and only if $E_P \subseteq E \cup S$, i.e., G_P is the subgraph of the digraph G extended with its loops S. In other words, a G-subordinated Markov kernel P is a stochastic matrix on V with support $E \cup S$. Then, the sum condition for a G-subordinated Markov kernel P can be rewritten as:

$$\sum_{w:v\to w} p_{vw} + p_{vv} = 1, \quad v \in V. \tag{1}$$

(Note that $p_{00} = 0$).

A p.d. $\boldsymbol{\pi}$ on V is a stationary distribution (s.d.) of the kernel P if $\Sigma_{u\in V} \pi_u p_{uv} = \boldsymbol{\pi}_v$ for all $v \in V$. For a G-subordinated Markov kernel P this formula, the so-called global balance equation, can be expressed as:

$$\sum_{u:u\to v} \pi_u p_{uv} + \pi_v p_{vv} = \pi_v, \quad v \in V. \tag{2}$$

Fig 3 presents a Markov kernel with its s.d. on the road network in Fig 1.

The stationary distribution can be derived by solving the linear Eq (2) numerically. Since the state space (the road network) is finite there exists at least one stationary distribution. However, in some cases, the stationary distribution is not uniquely defined by these equations.

We show that there is a direct connection between the uniqueness of s.d. of a Markov kernel P on V and the strongly connected property of the physical road network G if the Markov and graph structures are compatible with each other. The Markov kernel P on V is called G-compatible if, for any $u, v \in V$ such that $u \neq v$, $p_{uv} > 0$ if and only if $(u, v) \in E$. Note that the G-compatibility implies the weak G-subordination for a Markov kernel P, however the converse is not true.

Clearly, if P is G-compatible then the strong connectivity of G implies that the associated graph G_P to the Markov kernel P is also strongly connected. In this case, the Markov kernel

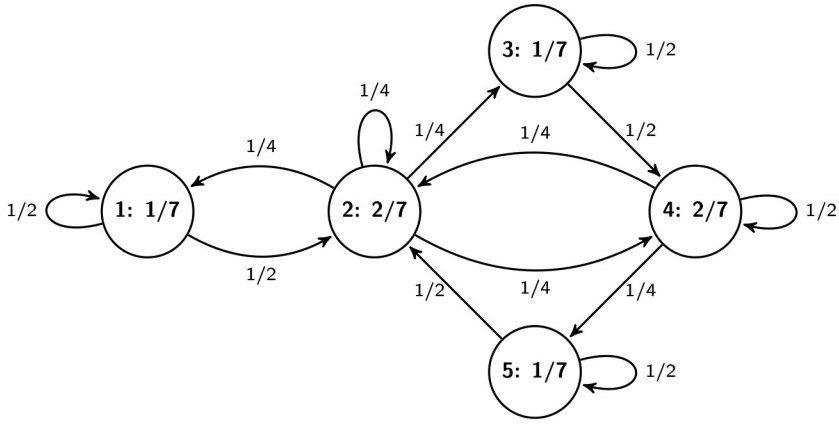


Fig 3. A Markov kernel (on edges) with its stationary distribution (on vertices) on the road network in Fig 1.

https://doi.org/10.1371/journal.pone.0246062.g003

(the transition matrix) P is called irreducible. Thus, by Theorem 1 in [64], see also Theorem 3.1 and 3.3 in Chapter 3 of [63] the following theorem holds.

Theorem 1. If a road network G is strongly connected then there is a unique stationary distribution $\boldsymbol{\pi}$ to any G-compatible Markov kernel P. Moreover, this distribution satisfies $\pi_v > 0$ for all $v \in V$.

The main consequence of this theorem is that, in case of any physical road network augmented by the ideal vertex 0, all of the Markov kernels defined on the road network that has positive transition probability on all roads have unique stationary distribution. Thus, it is reasonable to suppose that a real traffic which follows a Markovian dynamic has a local unique stationary distribution in a short time period that can be explored by observing the traffic.

Markov random walk and Markov traffic on road networks

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. Then a V-valued random variable (r.v.) is a $X: \Omega \to V$ measurable function, i.e., $X^{-1}(v) \in \mathcal{A}$ for all $v \in V$. In this case, X is a random function on the set V of vertices. For example, X can be the random position of a vehicle on the road network G, where the position refers to the actual vertex which the vehicle belongs to. Then, $\mathbb{P}(X^{-1}(v)) = \mathbb{P}(X = v)$ denotes the probability that a vehicle is at the vertex $v \in V$. By $\boldsymbol{\pi}_X(v) := \mathbb{P}(X = v)$, $v \in V$, a r.v. X induces a p.d. $\boldsymbol{\pi}_X$ on V.

A sequence $\{X_t\}_{t\in\mathbb{Z}_+}$ ($\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$) of V-valued r.v.'s is a Markov chain on the state space V if the Markov property holds:

$$\mathbb{P}(X_t = v_t | X_{t-1} = v_{t-1}, \ldots, X_0 = v_0) = \mathbb{P}(X_t = v_t | X_{t-1} = v_{t-1})$$

for all $t \in \mathbb{N}$, $v_0, \ldots, v_t \in V$. If X, X' are V-valued r.v.'s then for the conditional distribution $P = (p_{vv'})_{v,v'\in V}$, $p_{vv'} := \mathbb{P}(X = v | X' = v')$, $v, v' \in V$, we shall also use the notation $X|X'$. Clearly, $X|X'$ is a Markov kernel on V.

The main concepts of this paper are the Markov random walk and the Markov traffic defined in the following way.

Definition 2. Let the road network G be strongly connected and let P be a G-compatible Markov kernel on V with unique s.d. $\boldsymbol{\pi}$. Moreover, let $\{X_t\}_{t\in\mathbb{Z}_+}$ be a Markov chain on V such that $\boldsymbol{\pi}_{X_0} = \boldsymbol{\pi}$ and $X_t|X_{t-1} \sim P$ for all $t \in \mathbb{N}$. Then, $\{X_t\}_{t\in\mathbb{Z}_+}$ is called **Markov random walk** on the road network G with Markov kernel P.

The set $\{\{X_t^i\}_{t\in\mathbb{Z}_+}, i = 1, \ldots, k\}$ of k ($k \in \mathbb{N}$) mutually independent Markov random walks on G with Markov kernel P is called a **Markov traffic** of size k and it is parametrized by the quadruple $(G, P, \boldsymbol{\pi}, k)$.

The s.d. $\boldsymbol{\pi}$ of $\{X_t\}_{t\in\mathbb{Z}_+}$ can be considered as a categorical distribution (generalized Bernoulli distribution) on $\mathscr{F}$ by formula $\boldsymbol{\pi}(f) = \prod_{v\in V} \boldsymbol{\pi}_v^{f_v}$ where $\boldsymbol{f} \in \mathscr{F}$ is an indicator function, i.e., $f_v = 1$ for a fix $v \in V$ and 0 otherwise, see page 75 in [65]. Since the Markov traffic $\{\{X_t^i\}_{t\in\mathbb{Z}_+}, i = 1, \ldots, k\}$ consists of k mutually independent Markov random walks its s.d. becomes the k-fold convolution $\boldsymbol{\pi}^{*k}$ on $\mathscr{F}$, where * denotes the convolution. The p.d. $\boldsymbol{\pi}^{*k}$ fulfills

$$\boldsymbol{\pi}^{*k}(f) := k! \prod_{v\in V} \frac{\boldsymbol{\pi}_v^{f_v}}{f_v!} \tag{3}$$

where $\boldsymbol{f} \in \mathscr{F}$ is non-negative integer valued and satisfies the constraint $\sum_{v\in V} f_v = k$. In this case, $\boldsymbol{f}$ can be considered as traffic configuration where f_v counts the vehicles at vertex v. In fact, $\boldsymbol{\pi}^{*k}$ corresponds to the multinomial distribution with parameters k and $\boldsymbol{\pi}$, see Chapter 35 in [66]. Moreover, the r.v. $Y_t^v := \sum_{i=1}^k I(X_t^i = v)$ follows the binomial distribution with

parameters k and $\boldsymbol{\pi}_v$ for all $t \in \mathbb{Z}_+$ and $v \in V$, respectively. Note that the process $\{Y_t^v\}_{t\in\mathbb{Z}_+}$ denotes the (random) number of vehicles at vertex v in the Markov traffic.

A similar model to Markov random walk is proposed in [67] where binary-coded edge-valued r.v.'s are considered (as dual view) instead of our vertex-valued r.v.'s (as primal view). Note that if P is the uniform Markov kernel on G then we obtain the standard random walk of the graph theory, see the survey [68].

A Markov random walk is an individual Markov traffic with $k = 1$ in the sense that it describes the movement of a random vehicle which follows the stochastic rules defined by the Markov kernel. On the other hand, the Markov traffic provides a mathematical model for describing the traffic of k vehicles on a road network. Note that the independence assumption seems reasonable and not too strong because the vehicle controls are working separately form each other. For a pair $u, v \in V$ the notation $u \Rightarrow v$ will mean that $(u, v) \in E \cup S$, i.e., either $u \to v$ or $u = v$. If X_1, X_2 are random functions on V then $X_1 \Rightarrow X_2$ means that the two-dimensional distribution of (X_1, X_2) is concentrated on $E \cup S$. Clearly, for any Markov random walk $\{X_t\}_{t\in\mathbb{Z}_+}$ we have $X_t \Rightarrow X_{t+1} \Rightarrow \ldots \Rightarrow X_{t+n}$ for all t and $n \in \mathbb{N}$. One can also call $\{X_t\}_{t\in\mathbb{Z}_+}$ as a first-order random walk on the road network where a vehicle moves from vertex u to vertex v with probability p_{uv}. The second-order Markov random walk (traffic) on the line road network, where the vehicles move from edge to edge, can also be defined similarly, see [58].

Using the concept of two-dimensional s.d. a Markov traffic can be reparametrized in the following way. Introduce the two-dimensional distribution $Q = (q_{uv})$ on $V \times V$ as $q_{uv} \coloneqq \boldsymbol{\pi}_u p_{uv}$, $u, v \in V$. Then, Q is a two-dimensional s.d. on G in the following sense:

Definition 3. A matrix $Q = (q_{uv})_{u,v\in V}$ is called **two-dimensional stationary distribution** on G if (i) $q_{uv} \geq 0$ for all $u, v \in V$ and $q_{uv} = 0$ for all $u, v \in V$ such that $(u, v) \notin E \cup S$ (i.e., Q is weakly G-subordinated); (ii) $\Sigma_{u,v\in V}\, q_{uv} = 1$ (i.e., Q is a normalized matrix on V); and (iii) $\Sigma_{v\in V}\, q_{uv} = \Sigma_{v\in V}\, q_{vu}$ for all $u \in V$ (i.e., Q has equidistributed marginals).

A two-dimensional s.d. Q on G is called (strictly) positive if $q_{uv} > 0$ for all $u, v \in V$ such that $(u \Rightarrow v)\ u \to v$.

Property (iii) states that the two (row-wise and column-wise) marginal distributions of a two-dimensional s.d. on G coincide with each other. Clearly, for a Markov traffic, Q defined above is a positive two-dimensional distribution on G. Q can also be considered as a p.d. on the state space $E \cup S$, i.e., if we extend the set V' of vertices of $\mathrm{L}(G)$ as $V' = E \cup S$, on the line digraph. Thus, Q can be interpreted as the distribution of the vehicles on the edges of the road network, i.e., on the line digraph, see formula (11) in [21]. The distribution Q can also be visualized on the edges, see, Fig 4 for the simple example in Figs 1 and 11 in case of the Porto example discussed later. However, the converse of this statement is not true because there is p.d. on the line digraph which does not satisfy (iii). If $\{X_t\}_{t\in\mathbb{Z}_+}$ is a Markov random walk then the two dimensional distribution of any consecutive pair (X_t, X_{t+1}), $t \in \mathbb{Z}_+$, corresponds with Q.

Denote by $\mathscr{Q}$ the set of two-dimensional s.d. on G. One can easily see that $\mathscr{Q}$ is closed with respect to the affine combination. Namely, if $Q_1, Q_2 \in \mathscr{Q}$ then $\boldsymbol{\lambda}Q_1 + (1 - \boldsymbol{\lambda})Q_2 \in \mathscr{Q}$ for all $\lambda \in [0, 1]$.

Conversely, for a positive $Q \in \mathscr{Q}$, let us define

$$\begin{aligned} \pi_u &\coloneqq \sum_{v\in V} q_{uv} = \sum_{v\in V} q_{vu}, \quad u \in V, \\ p_{uv} &\coloneqq \frac{q_{uv}}{\pi_u}, \quad u, v \in V. \end{aligned} \tag{4}$$

Then, $P = (p_{uv})$ defines a G-compatible Markov kernel with s.d. $\boldsymbol{\pi} = (\pi_u)$ on G. Thus, a Markov

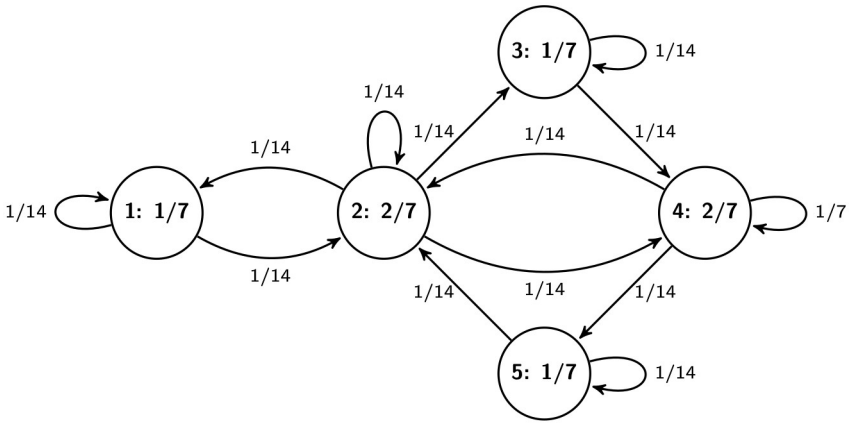


Fig 4. The two-dimensional s.d. (on edges) with its equidistributed marginals (on vertices) on the road network in Fig 1 for the Markov kernel in Fig 3. One can easily check that the sums of probabilities written on the edges in and out each vertex are equal, respectively.

https://doi.org/10.1371/journal.pone.0246062.g004

traffic defined by the quadruple $(G, P, \boldsymbol{\pi}, k)$ can be introduced by an equivalent way through the triplet (G, Q, k). However, it will turn out later that, from a statistical point of view, the parameter matrix Q can be estimated in a computationally more efficient way than the pair of transition matrix P and its s.d. $\boldsymbol{\pi}$.

Ergodicity of Markov traffic

The simulation method proposed in this paper is based on the ergodicity of Markov traffic which follows from the ergodic theory of finite Markov chains.

Let $\boldsymbol{\pi}_0$ be an initial distribution on V and define the n-th absolute p.d. $\boldsymbol{\pi}_n$ by the recursion $\boldsymbol{\pi}_n^\top = \boldsymbol{\pi}_{n-1}^\top P$, $n \in \mathbb{N}$. Clearly, $\boldsymbol{\pi}_n^\top = \boldsymbol{\pi}_0^\top P^n$, where the product of two Markov kernels P and Q on V is defined as $(PQ)_{uw} := \sum_{v \in V} p_{uv}\, q_{vw}$, $u, w \in V$. If G is strongly connected and we start from the unique s.d. $\boldsymbol{\pi}_0 = \boldsymbol{\pi}$ of P, see Theorem 1, then $\boldsymbol{\pi}_n^\top = \boldsymbol{\pi}^\top P^n = \boldsymbol{\pi}^\top$ for all $n \in \mathbb{N}$. Thus, in this case, $\boldsymbol{\pi}_n \to \boldsymbol{\pi}$ as $n \to \infty$. However, in general, the sequence $(\boldsymbol{\pi}_n)_{n \in \mathbb{N}}$ does not converge to the s.d. for all initial distribution $\boldsymbol{\pi}_0$ even if the s.d. is unique. However, if we consider the average of the n-th absolute p.d.'s in time, we have the convergence to the unique s.d. The Markov kernel which satisfies this property is called ergodic.

The following result is based on Theorem 4.1 in Chapter 3 of [63]. If a road network G is strongly connected then any G-compatible Markov kernel P is ergodic and the average Markov kernel A_n converges, i.e.,

$$A_n := (n+1)^{-1}(I + P + \ldots + P^n) \to \boldsymbol{\Pi} := \mathbf{1}\boldsymbol{\pi}^\top$$

as $n \to \infty$, where $\boldsymbol{\pi}$ is the unique s.d. of P. Moreover, the limiting probabilities of the time averages of the absolute p.d.'s satisfy

$$(n+1)^{-1}(\boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 + \ldots + \boldsymbol{\pi}_n) \to \boldsymbol{\pi} \tag{5}$$

as $n \to \infty$ for all initial p.d. $\boldsymbol{\pi}_0$.

In applications, along absolute p.d.'s, we may also be interested in some functionals of these distributions, e.g., the number of vehicles in a region of the road network. Define the functional $\boldsymbol{f}(\boldsymbol{\pi}) := \sum_{v \in V} f_v\, \pi_v$ of p.d. $\boldsymbol{\pi}$, where $\boldsymbol{f} \in \mathscr{F}$. Then, (5) can be extended that $n^{-1} \sum_{i=1}^n F(\boldsymbol{\pi}_i) \to F(\boldsymbol{\pi})$ as $n \to \infty$, see Theorem 4.1 of [63].

Instead of time averages, in order to achieve the convergence of n-th absolute p.d.'s we need the additional assumption of aperodicity for G, see Theorem 2.1 in Chapter 4 of [63]. If G is an aperiodic, strongly connected road network and P is a G-compatible Markov kernel on it, then the sequence of Markov kernels P^n, $n \in \mathbb{N}$, converges to the limiting Markov kernel $\mathbf{\Pi}$. Moreover, the limit of the sequence of n-th absolute p.d. $\boldsymbol{\pi}_n$ is the unique s.d. $\boldsymbol{\pi}$ to the Markov kernel P which is independent of the initial p.d. $\boldsymbol{\pi}_0$. For any functional F we also have that $\boldsymbol{F}(\pi_n)$ converges to $\boldsymbol{F}(\pi)$ as $n \to \infty$ on an aperiodic, strongly connected road network.

The ergodicity of Markov traffic with any G-compatible Markov kernel is derived in the same way. Let $\boldsymbol{\pi}_0^i$ and $\boldsymbol{\pi}_n^i$ denote the initial distribution and n-th absolute p.d., respectively, for the ith vehicle, $i = 1, \ldots, k$. Then, the ergodic property of Markov traffic, similarly to (5), can be formulated as

$$(n+1)^{-1}\sum_{i=0}^{n}(\boldsymbol{\pi}_i^1 * \ldots * \boldsymbol{\pi}_i^k) \to \boldsymbol{\pi}^{*k} \tag{6}$$

if $n \to \infty$, i.e., the time average of the probability of complex traffic events converges to a constant which corresponds to their stationary probability. Since the mean of the multinomial distribution with parameters k and $\boldsymbol{\pi}$ is calculated as $k\boldsymbol{\pi}$ one can see that the probabilities in s.d. $\boldsymbol{\pi}$ on V can be unfolded by the limit of state-space and time averages as

$$((n+1)k)^{-1}\sum_{i=0}^{n}\sum_{\boldsymbol{f}\in\mathscr{F}} f_v \cdot (\boldsymbol{\pi}_i^1 * \ldots * \boldsymbol{\pi}_i^n)(\boldsymbol{f}) \to \boldsymbol{\pi}_v \tag{7}$$

as $n \to \infty$ for all $v \in V$. Note that the left hand side of (7) is the average of the number of vehicles at vertex v in time divided by the size of the traffic (k).

The convergence results (6) and (7) guarantee that the unique s.d. of a G-compatible Markov kernel can be approximated and thus explored by long run behavior of the traffic flow on the road network. A visualization of the convergence of Markov traffic simulation to its s.d. is presented in Fig 10.

Trajectories from public datasets

For our experiments, we needed a dataset of real-life traffic trajectory data. In our terminology, a trajectory is a sequence of data that provides information about the path of a vehicle moving from a start to an end point, associating geographic coordinates with timestamps. We required a dataset that satisfies the following criteria:

1. Contains complete trajectories, i.e., the availability of only the start and end points is not sufficient, intermediate trajectory points must also be available.
2. The trajectory points must be sampled at a high enough frequency, so that the distance between consecutive points should not be too large, (e.g., an average distance of the order of 10 meters is acceptable, but an average distance of the order of 100 meters is definitely not).
3. The dataset is sufficiently large. It should cover a long enough period of time, preferably uniformly. The number of trajectories per day should be of the order of thousands.
4. Trajectories should cover a relatively small geographic area, e.g., a city or a district.
5. Vehicles should not follow a fixed route, e.g., public transport bus trajectories are not suitable.
6. Publicly available for research purposes.

These requirements were satisfied by the Taxi Trajectory Prediction (TTP) dataset from Kaggle. The dataset covers a period of one year from July 1, 2013 to June 30, 2014. It is split into a training and a test set, the former contains 1,710,670 trajectories, and the latter contains 320. The trajectories were collected in the city of Porto, Portugal, with a sampling rate of 15 seconds. First, we created a subset of the dataset, filtered to coordinates between W8.6518, W8.5771, N41.1129, N41.1756, see Fig 5. The data samples' features that were not relevant to the research, such as origin of call, identifiers for individual taxi or customers, and type of day (i.e. workday, weekend, holiday) were omitted. The processed format included the time of departure, both as a timestamp and as distinct date attributes, the length of the trajectory, and the points of the trajectory, represented as a list of GPS coordinates. Some data samples contained incomplete trajectories, these were discarded. Because of the properties of the proposed simulation model, the data was filtered to include only those samples that had a time of departure between 8-9 am. As a result, 82,345 trajectories remained. Although the length of trajectories had a wide range (the longest has 2,324 sample points), long trips were rare. Fig 6 shows the distribution of the length of trajectories. Most routes were around a length of 41 sample points, and routes with over 150 points were less than 1% of the dataset, see Fig 7. The distribution of the trajectory points (all, difference of start and end points, histogram of the difference) is shown in Fig 8. The descriptive statistics of the dataset is shown in Table 1.

Building graphs from OpenStreetMap data

OpenStreetMap (OSM) is a community project to build a free map of the world to which anyone can contribute. Data is available under the Open Data Commons Open Database License (ODbL). The representation and storing of map data is based on a simple but powerful model, that uses only three modeling primitives, namely, nodes, ways, and relations: 1. A node represents a geographical entity with GPS coordinates. 2. A way is an ordered list of at least two nodes. 3. A relation is an ordered list of nodes, ways, and/or relations. All of these modeling elements can have associated key-value pairs called tags that describe and refine the meaning of the element to which they belong. Users can export map data at the OSM web site manually, selecting a rectangular region of the map. Alternatively, map data can be extracted via web services, see http://wiki.openstreetmap.org/wiki/API. OSM uses two formats for exporting map data, namely OSM XML and PBF. Software libraries for parsing and working with OSM data are available for several programming languages, see https://wiki.openstreetmap.org/wiki/Frameworks.

We started our processing by building a graph from the OSM map of Porto, with the same bounding box as the filtered dataset. Specific nodes of the OSM file become the nodes in the graph. Because we only need those nodes that can be reached via vehicles, we had to filter the OSM file and collect only specific types of way nodes. In the OSM file, a way is a sequence of OSM nodes, so naturally, the nodes of ways become nodes in the graph. For every node we store the node's OSM ID, and its coordinates. We also insert an edge into the graph between every nodes in way. The weight of an edge is given by the squared distance between the nodes, which we calculate from the OSM file's data. We used pyosmium library for processing the OSM files and the NetworkX Python library for building the graph.

After building the graph we process the list of trajectories. Because the trajectories are given in GPS coordinates, we first have to translate those coordinates into OSM node IDs. For every coordinate in a trajectory, we search for the closest way node's coordinates in the built graph, so the result nodes have the same domain as the built graph's nodes. Obviously, the original trajectories made up of GPS coordinates does not have the same scaling as the OSM map. The coordinates in the trajectory are sampled in regular, but larger time intervals than the OSM, so

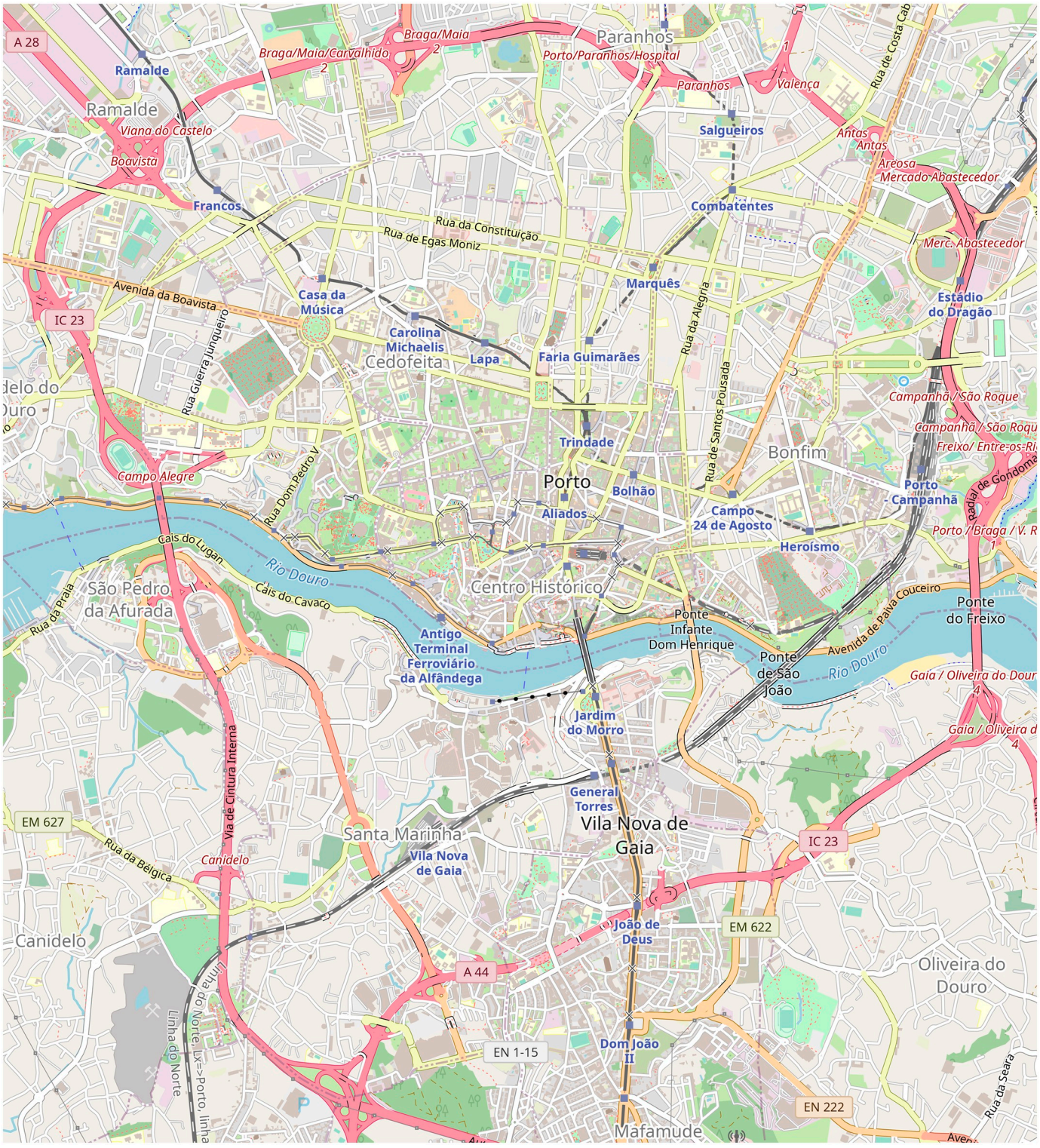


Fig 5. The map of the observed area. The graph created from the OSM data has 33,961 nodes, 53,126 edges, and covers a total of 857.26 km of road. The size of the area is about 43.68 km^2. (Base map and data from OpenStreetMap and OpenStreetMap Foundation. Reprinted from OpenStreetMap under a CC BY license, with permission from OpenStreetMap, original copyright 2020. ©OpenStreetMap contributors).

https://doi.org/10.1371/journal.pone.0246062.g005

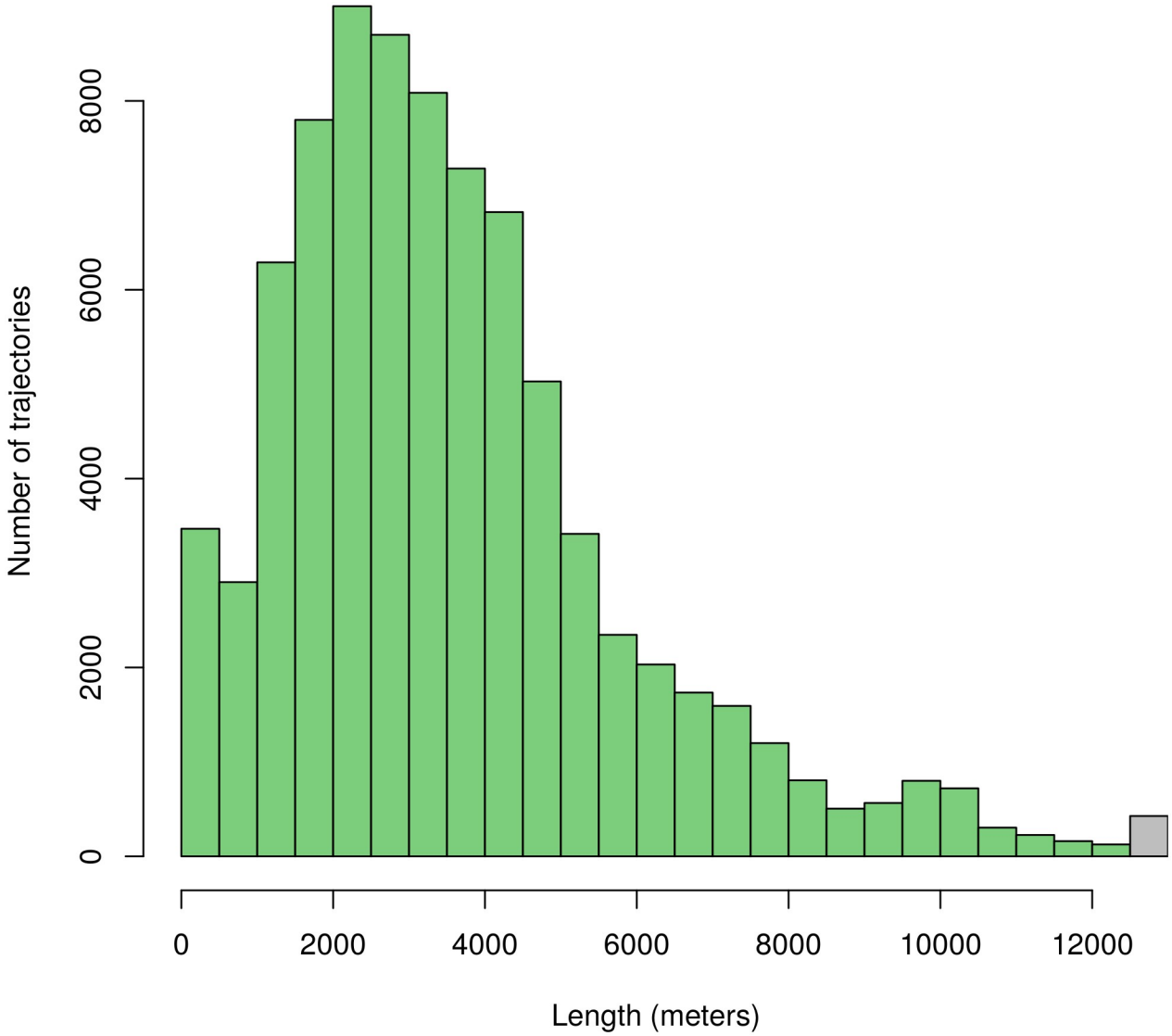


Fig 6. Histogram of trajectory lengths. The rightmost bar represents trajectories longer than 12,500 meters. The average trajectory length is 3,628.93 meters.

https://doi.org/10.1371/journal.pone.0246062.g006

they are not aligned. In order to match a trajectory to a way in our graph, we had to perform an interpolation on the result list of node IDs, so we ran a Dijkstra's shortest path algorithm on our graph between every node IDs for every trajectory. Because the OSM database contains errors, it can happen that in real life a route exists between two given places, but in the OSM database, there are no existing routes between those nodes that are representing the given places. In this case, we cut the faulty trajectories into pieces. The result of this process is an aperiodic strongly connected road network augmented by the ideal vertex 0, with a set of trajectories on the road network.

Statistical inference for Markov traffic using mobile sensors

The statistical analysis of a traffic systems described by the Markov traffic model means the estimation of the quadruple $(G, P, \boldsymbol{\pi}, k)$ or the triplet (G, Q, k) using observed data. To estimate

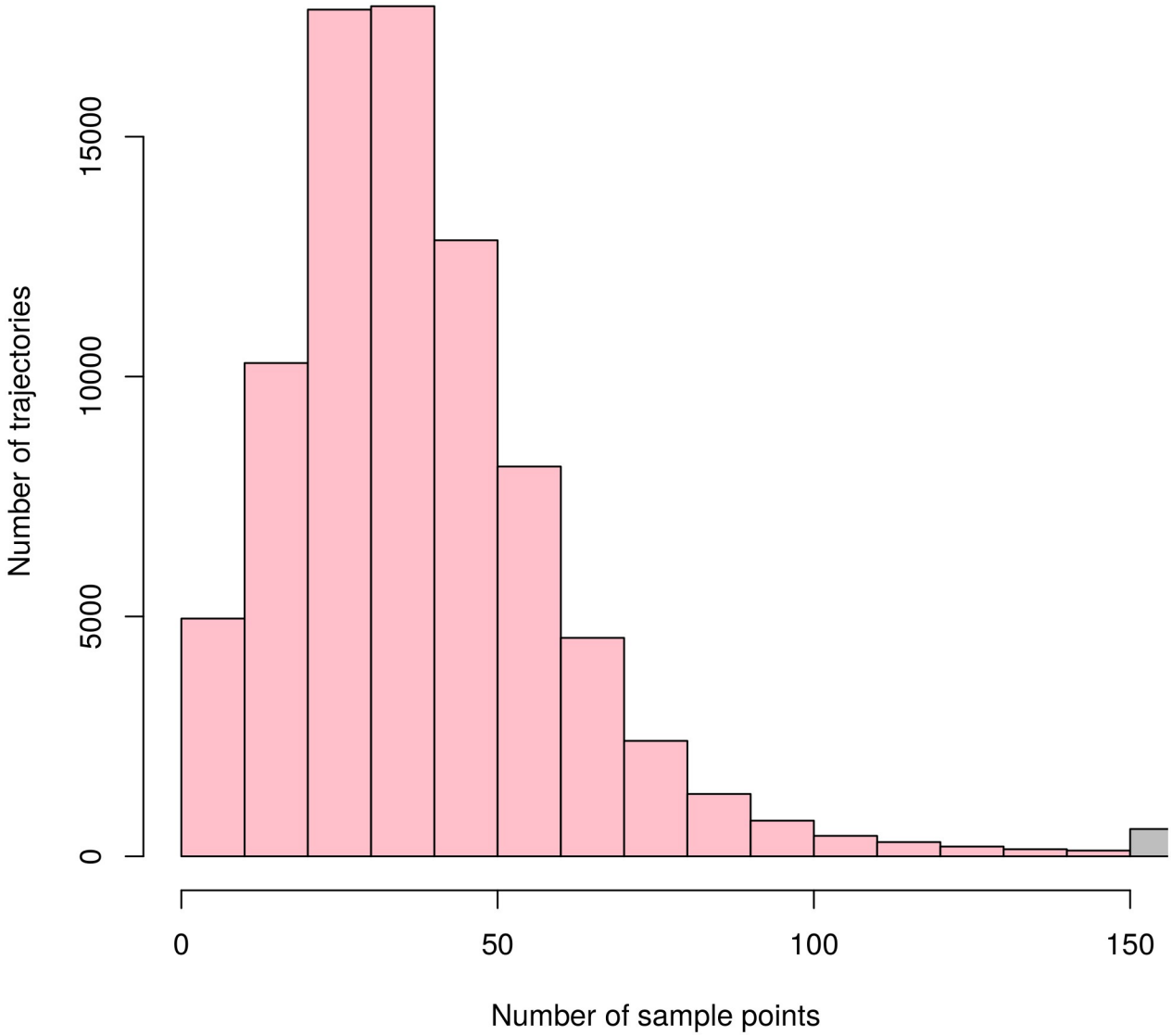
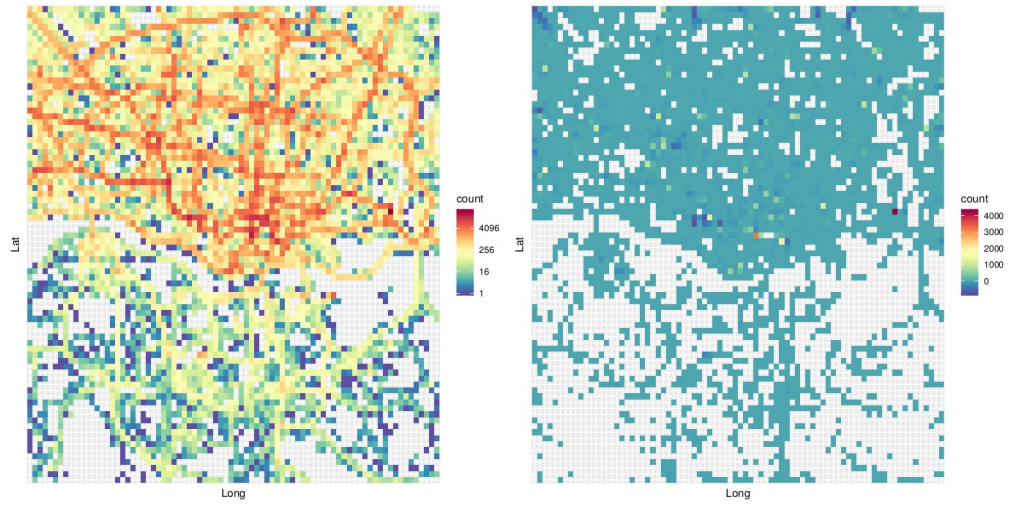


Fig 7. Histogram of number of sample points per trajectories. The rightmost bar represents trajectories with more than 200 sample points. On the average, a trajectory consists of 40 sample points and takes 10 minutes.

https://doi.org/10.1371/journal.pone.0246062.g007

G we have to explore the road system under study by identifying the set V of vertices and the set E of directed road segments. Fortunately, this exploring has already been done by a few organizations, see, e.g., the Google Maps and the OpenStreetMap. However, we should note that, in case of GPS-based trajectory data, we have to fit the data to the applied map system which is not an evident task at all. In the present paper, we propose a method for estimating the two-dimensional stationary distribution Q immediately instead of the pair $(P, \boldsymbol{\pi})$ of a transition matrix and its stationary distribution using mobile sensor data which may be gathered by vehicles, passengers etc. In this case, we have trajectories data which consists of the sequences of consecutive vertices, like in the TTP dataset. By (4), the estimators for P and $\boldsymbol{\pi}$ can be easily derived from an estimator of Q. Finally, it is supposed that the size k of the traffic is known.



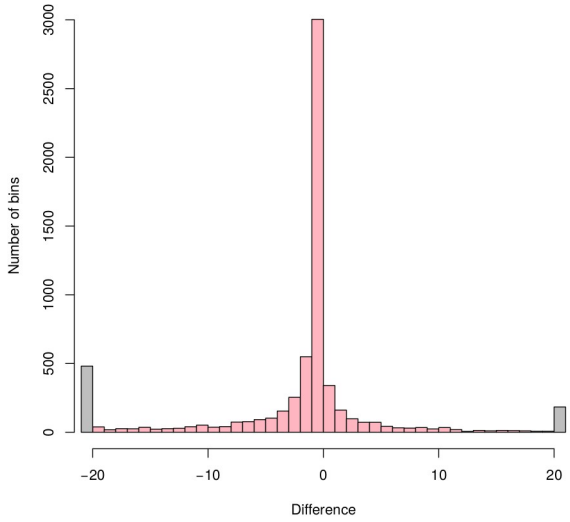


Fig 8. Distribution of trajectory points of the filtered dataset. a: Distribution of all trajectory points shown in a 2D histogram (number of bins: 80 × 80). b: Difference of trajectory starting and endpoints shown in a 2D histogram (number of bins: 80 × 80). The color of each bin represents the number of trajectory starting points minus the number of trajectory endpoints that fall in that bin. c: Histogram of the difference of trajectory starting and endpoints.

https://doi.org/10.1371/journal.pone.0246062.g008

Suppose that, for a Markov traffic, we observed a random sample of trajectories $\{X^i\}$, $i = 1, \ldots, k$, of size k defined by $X_1^i \Rightarrow X_2^i \Rightarrow \ldots \Rightarrow X_{n_i}^i$, $i = 1, \ldots, k$, where n_i denotes the length of the i-th trajectory. Let $n := n_1 + \ldots + n_k$ be the total sample size. Define the total two-dimensional consecutive empirical frequencies as:

$$n_{uv} := \sum_{i=1}^{k} n_{uv}^i, \tag{8}$$

Table 1. Descriptive statistics of lengths of trajectories. (82,345 total trajectories).

Name of statistics	Dist in points	Dist in meters
Mean	39.53	3,628.93
Median	35	3,176.786
Mode	34	0
Standard Deviation	31.64	2,408.93
Kurtosis	473.28	9.9
Skewness	12.11	1.78
Minimum	2	0
Maximum	2,324	61,055.58

https://doi.org/10.1371/journal.pone.0246062.t001

$u, v \in V$, where the trajectory-wise two-dimensional consecutive empirical frequencies, $i = 1, \ldots, k$, are defined as

$$n_{uv}^i := \sum_{j=1}^{n_i - 1} I(X_j^i = u, X_{j+1}^i = v),$$

$u, v \in V$. Plainly, n_{uv}^i denotes the number of consecutive (u, v) $(u, v \in V)$ pairs in the i-th trajectory. One can see that since $\{X^i\}$ is a proper Markov random walk we have $n_{uv}^i = 0$ for all $(u, v) \notin E \cup S$. Thus, the support of the two-dimensional frequency matrices $N := (n_{uv})_{u,v \in V}$, $N_i := (n_{uv}^i)_{u,v \in V}$, $i = 1, \ldots, k$, is a subset of $E \cup S$, i.e., they are weakly G-subordinated matrices. Clearly, $N = \sum_{i=1}^k N_i$ and we have

$$\sum_{u,v:u \Rightarrow v} n_{uv} = n - k, \tag{9}$$

where $n - k$ is the corrected sample size. Introduce

$$s_v := \sum_{i=1}^k I(X_1^i = v), \quad e_v := \sum_{i=1}^k I(X_{n_i}^i = v),$$

$v \in V$, i.e., s_v denotes the number of trajectories which start at vertex v and e_v denotes the number of trajectories which terminate at vertex v, respectively. Denote the one-dimensional marginal frequencies of N by $n_{v+} := \sum_{u \in V} n_{vu}$ and $n_{+v} := \sum_{u \in V} n_{uv}$, $v \in V$. We obtain that

$$n_{v+} + e_v = n_{+v} + s_v = n_v := \sum_{i=1}^k \sum_{j=1}^{n_i} I(X_j^i = v) \tag{10}$$

for all $v \in V$, where n_v denotes the number of vertex v in all trajectories. Finally,

$$\sum_{v \in V} s_v = \sum_{v \in V} e_v = k. \tag{11}$$

Define the vectors $\boldsymbol{s}$ and $\boldsymbol{e}$ on V as $\boldsymbol{s} := (s_v)_{v \in V}$ and $\boldsymbol{e} := (\boldsymbol{e}_v)_{v \in V}$, respectively. Then, (11) implies that $\mathbf{1}^\top (e - s) = 0$, i.e., the vectors $\boldsymbol{e} - \boldsymbol{s}$ and $\mathbf{1}$ are orthogonal.

The traditional maximum likelihood (ML) estimator $\widehat{P}_{\mathrm{ML}}$ of the transition matrix P is given by the maximization of the conditional loglikelihood

$$\log L = \sum_{u \Rightarrow v} n_{uv} \log p_{uv}$$

in parameters p_{uv}, $u, v \in V$ such that $u \Rightarrow v$, under the constraints $p_{u+} = 1$ for all $u \in V$. A solution of this constrained optimization problem is $\widehat{p}_{uv}^{\mathrm{ML}} = n_{uv}/n_{u+}$ for all $u, v \in V$ if $n_{u+} > 0$ and $\widehat{p}_{uv}^{\mathrm{ML}} = \delta_{uv}$ if $n_{u+} = 0$ where δ denotes the Kronecker delta. The maximum likelihood estimator $\widehat{\boldsymbol{\pi}}_{\mathrm{ML}}$ of the stationary distribution $\boldsymbol{\pi}$ is derived by the solution of the global balance equation $\boldsymbol{\pi}^{\top} = \boldsymbol{\pi}^{\top}\widehat{P}_{\mathrm{ML}}$ in $\boldsymbol{\pi}$. Thus, the maximum likelihood estimator $\widehat{Q}_{\mathrm{ML}} = (\widehat{q}_{uv}^{\mathrm{ML}})$ of Q is given by $\widehat{q}_{uv}^{\mathrm{ML}} = \widehat{\boldsymbol{\pi}}_{u}^{\mathrm{ML}}\widehat{p}_{uv}^{\mathrm{ML}}$, $u, v \in V$. In the sequel, a direct method is proposed for estimating the two-dimensional stationary distribution Q.

A naîve estimator for the two-dimensional stationary distribution Q based on the two-dimensional consecutive empirical frequency matrix N is $\widehat{Q}_{\mathrm{naïve}} := (n-k)^{-1}N$. Clearly, $\widehat{Q}_{\mathrm{naïve}}$, as a non-negative matrix on V, satisfies the properties (i) and (ii) of Definition 3. However, the problem with this naîve estimator is that its row and column marginals are not necessarily equal, i.e., in general, it does not satisfy the asumption (iii) of Definition 3. Hence, we have to introduce a new estimator $\widehat{Q}$ which belongs to $\mathscr{Q}$ and is optimal in some sense.

The optimality of the proposed estimator is defined by means of the least squares distance between matrices over G. Let $A = (a_{uv})_{u,v\in V}$ and $B = (b_{uv})_{u,v\in V}$ such that $a_{uv} = b_{uv} = 0$ for all $u, v \in V$ where $u \not\Rightarrow v$, i.e., let A and B be weakly G-subordinated matrices. The distance between A and B is defined as

$$\|A - B\|_G := \left(\sum_{u,v:u\Rightarrow v} |a_{uv} - b_{uv}|^2 \right)^{1/2}.$$

In fact, $\|\cdot\|_G$ is the Frobenius norm of the matrices of dimension $|V| \times |V|$ which vanish on the entries outside of $E \cup S$.

To formulate the objective function for estimating the two-dimensional stationary distribution it is convenient to weaken the assumptions of Definition 3 by leaving the normalizing assumption (ii). In the sequel, let $M = (m_{uv})$ denote a non-negative parameter matrix on G which satisfies assumptions (i) and (iii) of Definition 3, i.e., M is weakly G-subordinated and $\sum_{v\in V} m_{uv} = \sum_{v\in V} m_{vu}$ for all $u \in V$. Then, one can easily derive a two-dimensional stationary distribution Q from M by its normalization defining as $Q := (\mathbf{1}^{\top} M\, \mathbf{1})^{-1} M$.

Based on k number of trajectories, using the Frobenius norm, the optimality criterion is defined as the weighted sum of squared errors (SSE):

$$\mathrm{SSE}(M, \boldsymbol{w} \,|\, \boldsymbol{N}) := \sum_{i=1}^{k} w_i^{-1} \|N_i - w_i M\|_G^2, \tag{12}$$

where M is a non-negative parameter matrix satisfying assumptions (i) and (iii) of Definition 3, $\boldsymbol{w} = (w_i)_{i=1,\ldots,k}$ are non-negative unknown weights, with $\sum_{i=1}^{k} w_i = 1$, and $\boldsymbol{N} := (N_i)_{i=1,\ldots,k}$ denotes the data, where N_i is the two-dimensional consecutive empirical frequency matrix for the ith trajectory, see (8). The statistical inference for a Markov traffic means the minimization of the objective function SSE in its parameters M and $\boldsymbol{w}$ deriving the weighted least squares (WLS) estimators $\widehat{M}_{\mathrm{WLS}}$ and $\widehat{\boldsymbol{w}}_{\mathrm{WLS}}$. Then, the WLS estimator of Q is defined as $\widehat{Q}_{\mathrm{WLS}} := n_{\mathrm{eff}}^{-1}\widehat{M}_{\mathrm{WLS}}$ where $n_{\mathrm{eff}} := (\mathbf{1}^{\top}\widehat{M}_{\mathrm{WLS}}\mathbf{1})$ is the so-called effective sample size. Here, $\widehat{Q}$ can be interpreted as the estimated two-dimensional stationary distribution which describes the individual Markov traffic in time. On the other hand, n_{eff} gives the equivalent sample size related to the independent case which may be thought of as the information content of the observed data. Note that n_{eff} is not necessarily an integer and is different from n and $n - k$. Finally, w_i gives the importance of the ith trajectory in the sample. One can see that longer trajectory implies higher weight.

To formulate our result on WLS estimation of Markov traffic we need some basic facts on the spectral theory of directed graphs, see [69] for details. The symmetric unnormalized graph Laplacian matrix L of a digraph G is defined as

$$L := D - A - A^{\top}$$

where A denotes the adjacency matrix of G and $D := \mathrm{diag}\{\boldsymbol{d}^{+} + \boldsymbol{d}^{-}\}$ is a diagonal matrix. Note that for the road graph G, since there is no loop, we have $l_{vv} = d_{vv} = \deg^{+}(v) + \deg^{-}(v)$ for all $v \in V$. The main theorem of this paper is the following.

Theorem 2. There is a unique pair $(\widehat{M}_{\mathrm{WLS}}, \widehat{\boldsymbol{w}}_{\mathrm{WLS}})$ which minimizes the weighted sum of squared errors SSE defined in (12). These WLS estimators are derived as

$$\widehat{w}^{i}_{\mathrm{WLS}} := \frac{\|N_i\|_G}{\sum_{j=1}^{k} \|N_j\|_G},$$

$i = 1, \ldots, k$, and

$$\widehat{M}_{\mathrm{WLS}} := N + (\mathbf{1}\boldsymbol{\lambda}^{\top} - \boldsymbol{\lambda}\mathbf{1}^{\top}) \circ A,$$

where $\boldsymbol{\lambda} \in \mathscr{F}$ is called Lagrange vector and defined as a unique solution to the linear equation $L\,\boldsymbol{\lambda} = s - \boldsymbol{e}$ which satisfies the constraint $\mathbf{1}^{\top}\boldsymbol{\lambda} = 0$ (i.e., $\sum_{v \in V} \lambda_v = 0$) and $\circ$ denotes the entrywise (Hadamard) product of matrices.

Based on the previous theorem, by (9), the effective sample size is given as

$$n_{\mathrm{eff}} := (n - k) + (\boldsymbol{d}^{-} - \boldsymbol{d}^{+})^{\top}\boldsymbol{\lambda}, \tag{13}$$

i.e., n_{eff} depends only on the graph structure of the road network, which is independent of the data, the traffic direction vector $\boldsymbol{s} - \boldsymbol{e}$, and the corrected sample size. However, it does not depend on the data which are inside the trajectories.

The WLS estimators proposed above can be considered as a kind of composite (or quasi-) likelihood estimators for Markov chains, see [54]. The composite likelihood method is widely applied in complex statistical models when the full ML method can be difficult to apply or may not be robust enough. In our method, the objective function is based on pairwise marginal distributions, however, instead of formula (2) in [54], the quasi-likelihood function is a square function, the logarithm of the normal probability density with heteroscedastic variance which depends on the length of trajectories. The latter will be more clear by introducing the mean squared error (MSE) as

$$\mathrm{MSE} := n_{\mathrm{eff}}^{-1}\mathrm{SSE} = \sum_{i=1}^{k} n^{i}_{\mathrm{eff}} \|(n^{i}_{\mathrm{eff}})^{-1} N_i - Q\|^2_G, \tag{14}$$

where $n^{i}_{\mathrm{eff}} := w_i n_{\mathrm{eff}}$ denotes the effective sample size of the ith trajectory, $i = 1, \ldots, k$. The parameters of the objective function MSE are the effective sample sizes $\{n^{i}_{\mathrm{eff}}\}$ and the two-dimensional s.d. Q. The heuristic explanation of the need to use weights in formulas (12) and (14) is the following. By the Central Limit Theorem, for large n_i, the trajectory-wise two-dimensional consecutive empirical frequency matrix N_i can be approximated as $N_i \approx n_i Q + n_i^{1/2}\boldsymbol{\xi}_i$, where $\boldsymbol{\xi}_i$ is a normally distributed random matrix on V for all $i = 1, \ldots, k$, which is a heteroscedastic equation between the observed N_i and the parameter matrix Q. Hence, $\|N_i - n_i Q\|^2_G \approx n_i \|\boldsymbol{\xi}_i\|^2_G$, where $\|\boldsymbol{\xi}_i\|^2_G$, $i = 1, \ldots, k$, are independent identically distributed r.v.'s. Thus, we have to normalize the trajectory-wise squared errors proportionally to their lengths, respectively, in order to get balanced error terms.

The estimation theory of finite Markov chains goes back for a long time, see [70]. In the traditional ML approach the estimators of the transition and stationary probabilities are derived by corresponding relative frequencies, respectively. However, these estimators have a few problems which imply that they can be applied with limited success for estimating the Markov traffic on a road network. Firstly, they are based on only one long trajectory (or realization). However, in a real traffic dataset there is a large number of relatively short trajectories, i.e., the set $\{n_i, i = 1, \ldots, k\}$ are bounded, where k is large or tends to infinity. In our example, for the TTP dataset, the number of trajectories is above 80K with the mean length 40 and maximum length 2K, see Table 1. Secondly, they are asymptotic estimators in the sense that, for finite sample size, the estimated stationary distribution does not satisfy the global balance equation given by the estimated transition probability matrix. The global balance equation holds only asymptotically, i.e., when the sample size tends to infinity. In fact, the inaccuracy in the global balance equation is not too large, however, this little bias can cause significant discrepancy from the "true" stationary distribution in the simulation. Thirdly, the trajectories are biased during a short time period in the sense that they are starting from some parts of the road network and ending at other parts. For example, in the morning period the vehicles are moving from the residential districts to the business districts of the city and they are moving back in the afternoon period. In other words, the traffic has a definite direction on the road network. To demonstrate this behavior in the case of TTP dataset, Fig 8c shows the distribution of the elements of the traffic direction vector $\boldsymbol{s} - \boldsymbol{e}$ while Fig 8b shows their spatial distribution. Neither distributions are concentrated around the zero. The known improvements of the ML estimators, e.g., by using the bootstrap, see [71], do not solve these problems. However, the WLS estimator of the two-dimensional stationary distribution proposed in this paper is able to handle all of these problems. The estimator $\widehat{Q}_{\mathrm{WLS}}$ is taking account of more than one trajectory with their length. It determines uniquely both the transition probability matrix and its stationary distribution by (4) which satisfy the global balance equation obviously. Finally, by taking account of the traffic direction vector in the estimator, it can correct the bias due to the unbalanced sampling of trajectories on the road network.

The fundamental statement of Theorem 2, as one of the main result of this paper, is that the estimator $\widehat{Q}_{\mathrm{WLS}}$ (or $\widehat{M}_{\mathrm{WLS}}$) consists of two parts: the first part is the naîve estimator for the distribution of the consecutive pairs in trajectories based on the empirical frequencies, while the second part is a correction term ensuring that $\widehat{Q}_{\mathrm{WLS}}$ (or $\widehat{M}_{\mathrm{WLS}}$) has equidistributed marginals. The second part also depends on two components. The first one is the Laplacian matrix of the road graph which depends only on the graph structure of the road network and independent from the trajectory data. The second one is the traffic direction vector which depends only on the trajectory data. Note that all sufficient statistics, namely the total two-dimensional consecutive empirical frequencies and starting and ending empirical frequencies, can be computed by counting, which is numerically very effective and can be executed even for big data.

The computationally intensive part of WLS estimator is the numerical solution of sparse linear equation system given for the Lagrange vector $\boldsymbol{\lambda}$ in Theorem 2. This can be performed in a numerically effective manner by the eigenvalues-eigenvectors decomposition of the symmetric unnormalized (or normalized) graph Laplacian matrix, see Proposition 1 and 2 in [72]. Remark that the first few significant eigenvalues and eigenvectors, being independent from the data, can be computed and stored in advance for a simulation program. Finally, one can also see that, similarly to the Google's PageRank algorithm, see Chapter 15 in [73]), a linear recursion could be computationally more efficient in large-scale problems.

Results

To evaluate the performance of the proposed WLS estimation method by comparing it to the traditional ML one discussed above, a simple simulation study was conducted at different sample sizes for small and medium road network. In the simulations, in order to mimic the real traffic, we tried to keep the length of trajectories low and the number of trajectories high compared to the size of the road network, similarly to the Porto example. The absolute bias of an investigated estimator $\widehat{Q}$ for the two-dimensional s.d. Q as a parameter is defined by $\|\widehat{Q}-Q\|_G$. The empirical absolute bias and its standard error (SE) correspond to the mean and standard deviation of absolute biases in 100 replications, respectively. All simulations were carried out in Python using the PyDTMC library developed for analysing discrete time Markov chains (https://pypi.org/project/PyDTMC/). The codes and datasets of our simulation are available upon request.

Table 2 displays the simulation results for the small road network in Fig 1 using the Markov kernel of Fig 3 (We implemented this example in Python, see: https://github.com/rbesenczi/Crowd-sourced-Traffic-Simulator/blob/master/model-sources/Markovkernel/example_graph.py). The simulation parameters were k = 100, 200, 500, and 1000 number of trajectories with n = 3, 5, and 10 length. The absolute bias and its standard error do not depend on the length n and they are decreasing as k is increasing for both estimation methods. The latter is an expected result. Moreover, while for relatively small k the performance of the WLS and ML methods are similar, in the case of relatively large number of trajectories the ML estimator outperforms the WLS one a little bit. This phenomenon could be due to the asymptotic optimality of the ML estimator because the parameter k is enough large (1000) compared to the size of the road graph (5).

In the second simulation scenario, a strongly connected subgraph, which contains 1000 vertices, of Porto's road network was chosen (exported from the OSM, as well, GPS coordinates W8.6137, W8.5991, N411573, N41.1437). The entries of Markov kernel were generated randomly. The simulation parameters were k = 1000, 3000, and 5000 with n = 3, 5, and 10. In this scenario, the absolute bias and its standard error depend also only on k and are independent of the length n. However, there are significant differences between the performances of the two estimation methods (ML and WLS) related to the parameter k. On the one hand, the absolute bias of ML estimator is decreasing as k is increasing while it is constant for WLS estimator. On the other hand, the WLS estimator is better than the ML one in case of k = 1000 but worse in

Table 2. Simulation results, absolute bias and SE (inside parenthesis), for the Markov kernel in Fig 3 on the road network in Fig 1. (k—number, n—length of trajectories).

k	n	ML	WLS
100	3	0.034 (0.0103)	0.034 (0.0109)
100	5	0.035 (0.0106)	0.034 (0.0103)
100	10	0.033 (0.0095)	0.033 (0.0094)
200	3	0.024 (0.0066)	0.026 (0.0066)
200	5	0.023 (0.0064)	0.024 (0.0067)
200	10	0.024 (0.0071)	0.025 (0.0070)
500	3	0.015 (0.0046)	0.017 (0.0049)
500	5	0.015 (0.0041)	0.017 (0.0049)
500	10	0.016 (0.0047)	0.017 (0.0049)
1000	3	0.010 (0.0032)	0.013 (0.0041)
1000	5	0.011 (0.0034)	0.015 (0.0044)
1000	10	0.010 (0.0030)	0.014 (0.0040)

https://doi.org/10.1371/journal.pone.0246062.t002

Table 3. Simulation results, absolute bias and SE (inside parenthesis), for a part of Porto's map with 1000 vertices. (k—number, n—length of trajectories).

k	n	ML	WLS
1000	3	0.166 (0.0559)	0.025 (0.0007)
1000	5	0.184 (0.1214)	0.025 (0.0007)
1000	10	0.169 (0.0938)	0.025 (0.0008)
3000	3	0.064 (0.1725)	0.023 (0.0005)
3000	5	0.070 (0.1665)	0.023 (0.0005)
3000	10	0.063 (0.1705)	0.023 (0.0005)
5000	3	0.016 (0.0150)	0.023 (0.0004)
5000	5	0.014 (0.0055)	0.023 (0.0004)
5000	10	0.014 (0.0126)	0.023 (0.0003)

https://doi.org/10.1371/journal.pone.0246062.t003

case of $k = 5000$. Since the former parameter setting is closer to the real traffic, this simulation corroborates the superiority of WLS method based on two-dimensional s.d. against the traditional maximum likelihood. Finally, in this scenario, the scale of the SE's indicates that the WLS estimator is more stable than the ML one See Table 3.

We have also implemented the model in the OOCWC system in order to apply our simulation method for real large-scale problems. First, we have filtered the TTP dataset. Then, we have created the Markov kernel from the filtered dataset, so all nodes of the simulation graph will have the corresponding transition probability vector. We should note, however, that not all nodes can be found in the Markov kernel, because it can happen that the dataset does not completely cover the whole map, i.e., not all nodes are part of a trajectory. In this case, we set uniform distribution for the corresponding node. Finally, we had to modify the basic operation of the simulation algorithm. In the original implementation, the cars are moving on the map quite randomly. Now, a car selects the next node based on the transition probability vector of the current node. For this, we use the pseudo-random number generation engine from the Boost Random library that is based on the method presented in paper [74].

Let's consider an example. We are at the graph vertex (or intersection) of OSM node ID 1110673569 (with GPS coordinates 41.1752185, -8.6231927). The total transitions of this node (i.e. the total trajectories that cross this intersection) in the dataset is 1,649. The transitions to the neighbor nodes are shown in Table 4. Please note that the actual transition probability (TP) is not the same as the ratio of the transitions to the neighbor node and the total transitions of the node which is called frequency (or ML) based transition probabilities. The actual transition probability comes from the Markov kernel of the whole graph. The two kinds of transition probabilities are also compared in Table 4 where the WLS based transition probabilities have been derived by our method. One can already see in this simple example that the difference between the two methods could be huge. This small example can be observed in Fig 9, as well.

Table 4. Transitions of intersection 1110673569. (TP—transition probability).

Neighbor node	# of transitions	ML based TP	WLS based TP
1471136241	1449	0.879	0.6
1110673512	170	0.103	0.382
1837918561	30	0.018	0.018
Sum	1649	1	1

https://doi.org/10.1371/journal.pone.0246062.t004

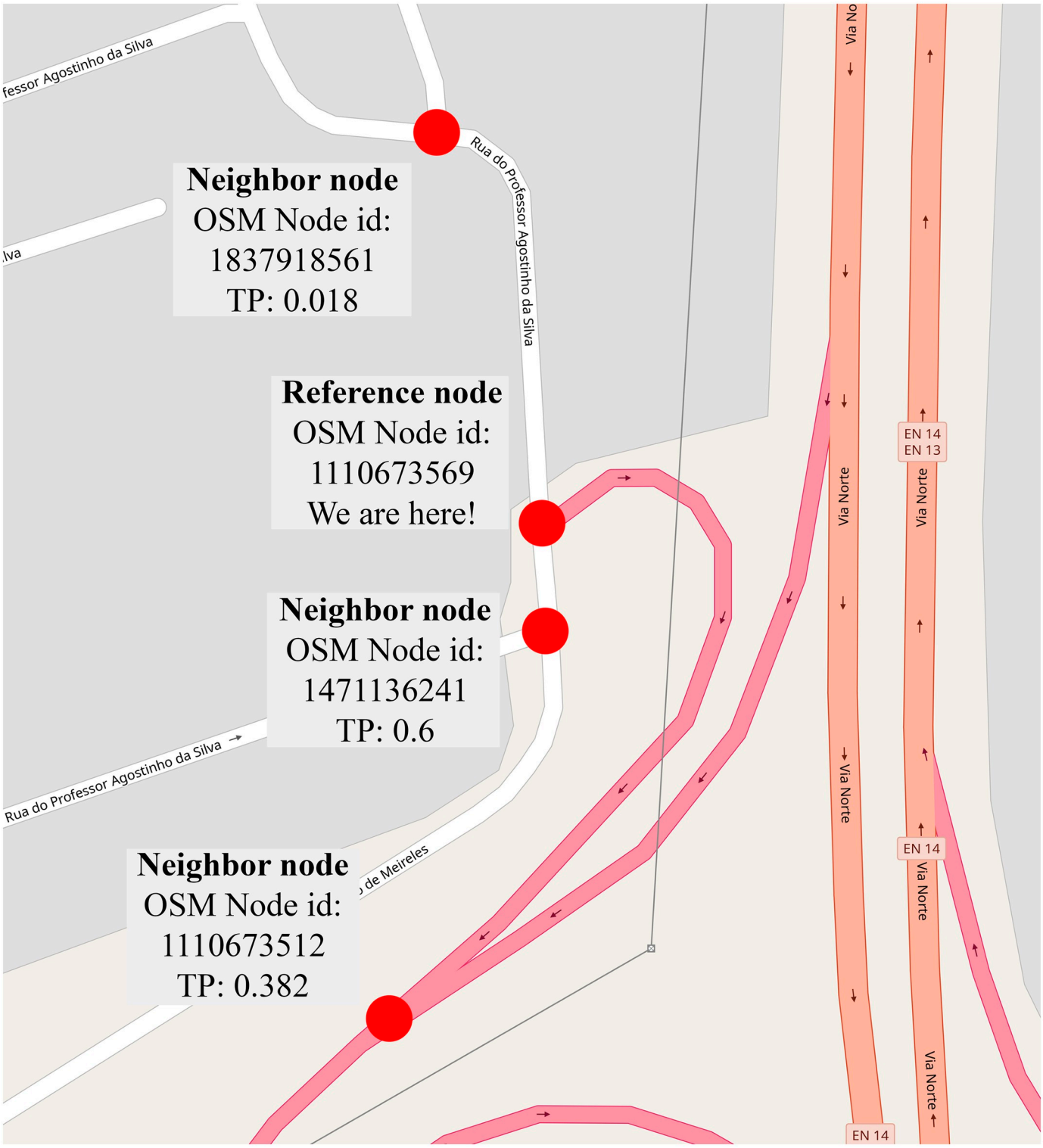


Fig 9. A visual explanation of transitions of intersection 1110673569. TP means transition probability, red dots indicate nodes. (Base map and data from OpenStreetMap and OpenStreetMap Foundation. Reprinted from OpenStreetMap under a CC BY license, with permission from OpenStreetMap, original copyright 2020. ©OpenStreetMap contributors. Annotated by the authors).

https://doi.org/10.1371/journal.pone.0246062.g009

The initialization phase of the simulation adds traffic units to the map. Each unit is placed to an OSM node, i.e., on a vertex of the simulation graph. There exist two ways to do this, one is following a prescribed distribution (e.g. uniform), the other is following measured data. In our test case, we initialized simulations with fictional measured data. We put units only to the streets Rua de Antero de Quental, Rua da Constituição and Rua da Boavista (25.6%, 51.4%, 23% of the cars, respectively), i.e., the simulation starts from the traffic configuration which is concentrated on three nodes of the road graph. In addition, we can set the number of the simulation units. We run simulations with k = 5, 000, 10, 000, 20, 000, 30, 000 and 50, 000 units. The simulation starts when all simulation units are added to the map. Fig 10 shows the change of the distribution of cars during the simulation.

The RCE produces a logfile that contains the position of every simulation unit in every simulation step. From this file, we calculate the number of cars by streets in every minute, so we can observe the change of distribution of the cars. In addition, we calculated the s.d. of cars for streets in the city of Porto, see Fig 11. This latter one tells us, what is the probability that a car is on a given street. It is worth noting the similarity between this figure and Fig 8a. The ticker line on Fig 11 corresponds to increasingly hot color on Fig 8a.

To obtain a quantitative measure that describes the "goodness" of our simulation algorithm, we applied the Pearson's chi-squared test. We expect, by the ergodicity of the Markov traffic, that during the simulation, independently from the initial distribution, within a certain time period, the distribution of the cars become close to the previously calculated stationary distribution. Fig 12 shows the test results. We can observe that in the first few minutes the test statistic is significantly high, meaning that the distribution of the cars is still far from the steady-state. However, after a time period that depends on the number of cars, the test statistic becomes low, meaning that the distribution became steady. One can observe that it takes more time to reach the steady-state with more traffic units, which is reasonable. Another notable trend is the case of 5,000 cars, where the line is elevating after reaching the steady-state. This can be caused by the low number of cars. The number of individual streets (named or unnamed, e.g. motorway junctions) is 2,194. 5,000 cars are simply not enough to reach and hold a steady-state in this type of simulation.

Finally, we should note some implementation details and possible drawbacks of this model that may have an impact on the model's overall performance.

In small and medium graphs, the proposed algorithm and its implementation performs as it is expected. But in the case of our Porto example, where the graph has 33,961 nodes and 53,126 edges and the TP matrix is very sparse, numerical problems may occur. One problem can occur when we calculate the $\widehat{Q}_{\mathrm{WLS}}$ estimator and then TP matrix. For matrices with this size (34,000 x 34,000), we cannot solve the linear equation of the Lagrange vector in Theorem 2 always numerically, thus we could only use the least square solution for a numerically stable calculation. In some cases, this causes impossible numbers to present in the TP matrix, e.g. for a node, the TP vector is [1.17489, -0.174894], which is obviously impossible. It is interesting to note that the sum of these "malfunctional" TP vectors are 1 all the time, and mostly occurs if the node has a low number of transitions (less than 20). In such cases, we use the frequency based TP. Another numerical problem can occur when we calculate the s.d. $\boldsymbol{\pi}$, namely, negative values may present in the results. We need to handle this problem when we calculate the Pearson's chi-squared test. We chose to shift every value of $\boldsymbol{\pi}$ until we get a sum of 1 for $\boldsymbol{\pi}$.

Some minor issues can occur with the map database and the differences between the Porto dataset and the OSM data. In some cases, we could not calculate a route between two consecutive trajectory points using OSM data. This can happen because of the imperfection of the

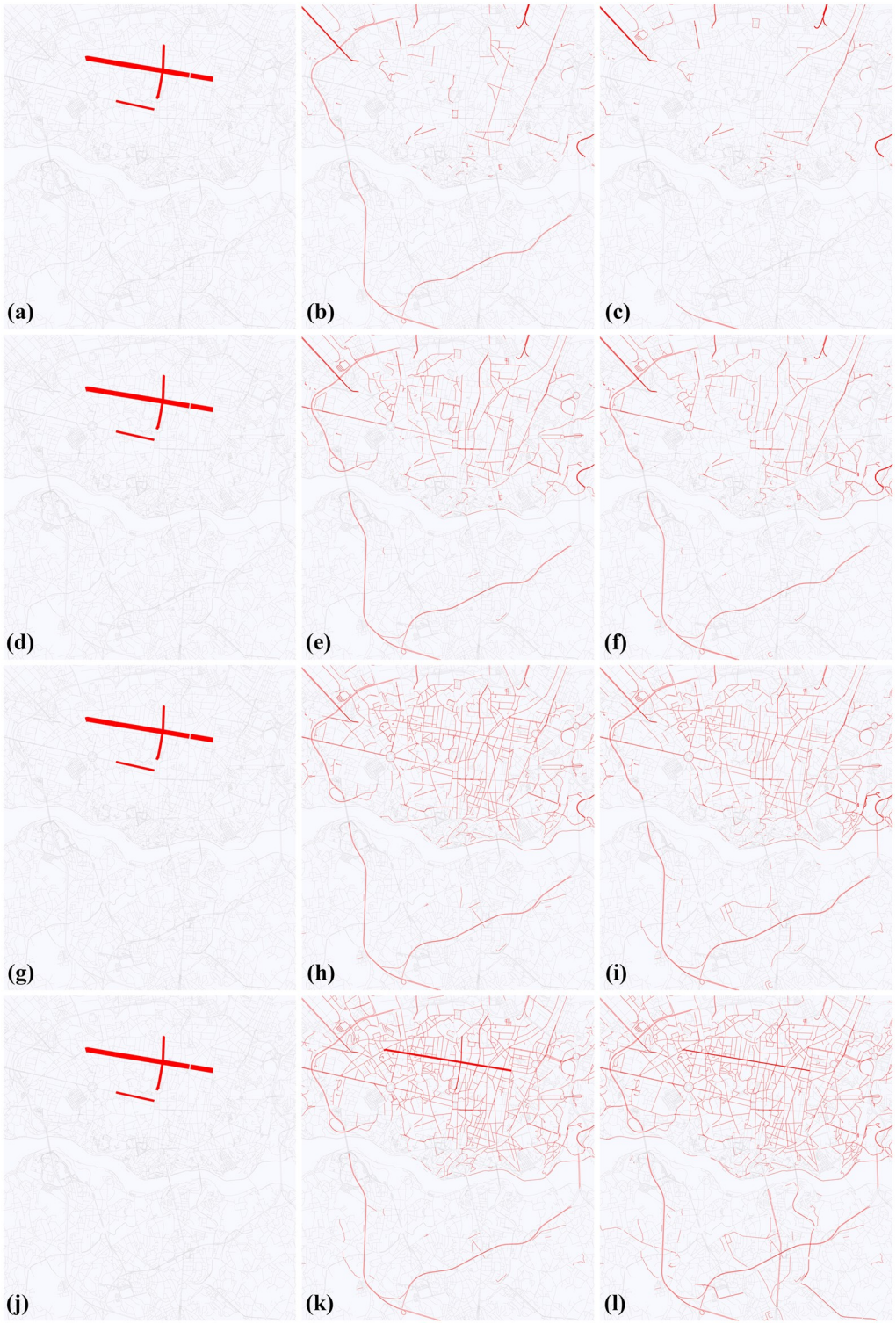


Fig 10. The change of the distribution of cars during the simulation (5,000, 10,000, 20,000 and 50,000 cars). The thickness of the street is proportionate with the number of cars on the street. a: Initial step (5,000 cars). b: After 30 mins (5,000 cars). c: After 60 mins (5,000 cars). d: Initial step (10,000 cars). e: After 30 mins (10,000 cars). f: After 60 mins (10,000 cars). g: Initial step (20,000 cars). h: After 30 mins (20,000 cars). i: After 60 mins (20,000 cars). j: Initial step (50,000 cars). k: After 30 mins (50,000 cars). l: After 60 mins (50,000 cars).

https://doi.org/10.1371/journal.pone.0246062.g010

Fig 11. The stationary distribution of cars in Porto based on the TTP dataset.

https://doi.org/10.1371/journal.pone.0246062.g011

OSM data or false GPS measurement. We handle this case by splitting the trajectory into pieces.

Another minor issue arises in the calculation of the Pearson's chi-squared test. Since the OSM Porto map and the trajectory dataset do not cover each other perfectly, we only know the s.d. $\boldsymbol{\pi}$ for a subgraph of the whole map. During the simulation the units can traverse the whole map graph, so, it can happen that a traffic unit reaches an edge which is not part of the subgraph where we know the s.d. $\boldsymbol{\pi}$. During the calculation of the Pearson's chi-squared test, we consider only those cars that are present on the road network, where the s.d. $\boldsymbol{\pi}$ is known.

Conclusions

In this paper, we have described our traffic simulation model that is called "Markov traffic" based on tools from graph theory and Markov modeling. The aim was to provide a simulation

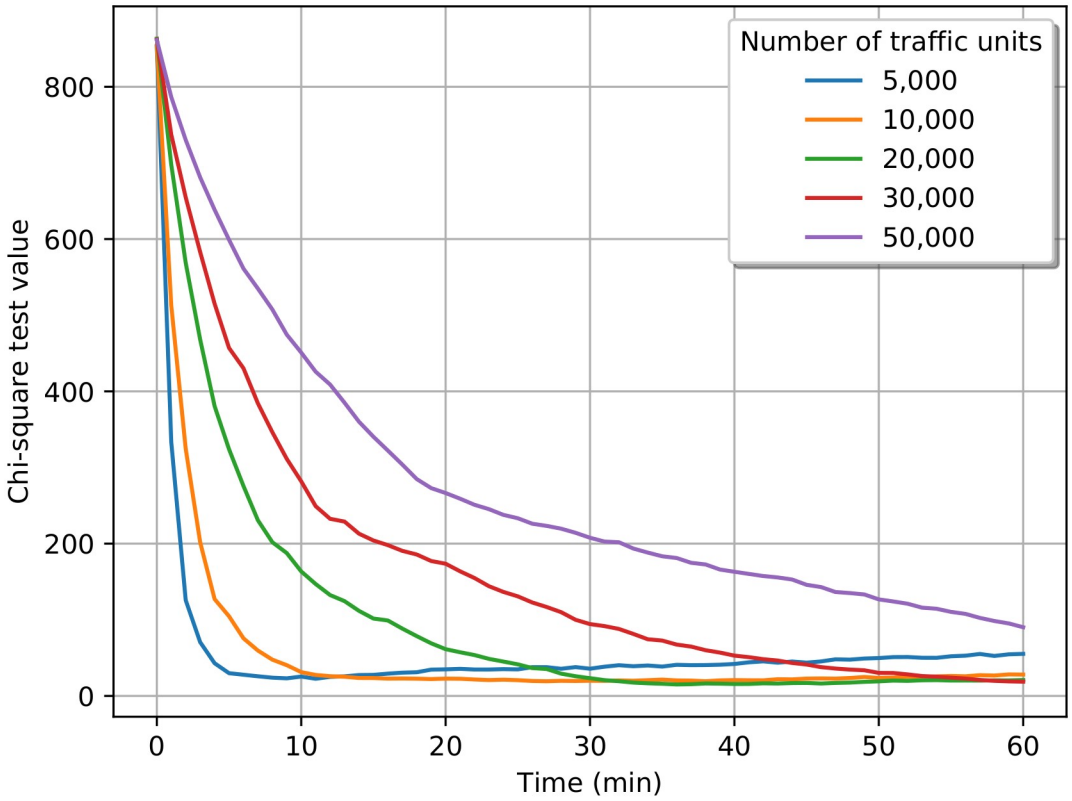


Fig 12. Chi-square test results.

https://doi.org/10.1371/journal.pone.0246062.g012

method that is able to keep the distribution of the cars on the map in a steady-state on a large scale road network. We have proven that, under general assumptions, the stationary distribution (s.d.) is unique for any Markov transition mechanism on a wide class of road networks. An explicit formula has also been derived for the s.d. and the ergodicity of Markov traffic has also been proved.

We have shown that the s.d., with the related transition mechanism, can be explored from observed data based on sample trajectories. We have provided a statistical method and proved its optimality by simulation with which we can create the Markov kernel necessary to obtain a Markov traffic on a given road graph. Using this kernel, we can initiate traffic simulations that provide a s.d. of the cars on the map.

To provide an example for creating this kernel file, we have used a publicly available dataset, namely the Taxi Trajectory Prediction dataset. Our simulation uses OpenStreetMap, a free map database.

To test our theories, we have implemented the proposed model in our simulation program (RCE). We have run simulations and it has been proved to provide a s.d. on the map graph of Porto, Portugal. The whole project (including the RCE) is available for download (see https://github.com/rbesenczi/Crowd-sourced-Traffic-Simulator/blob/master/justine/install.txt). Some simulation video is available at the YouTube channel of the first author at http://bit.ly/2FRpPxL.

Future work will focus on the further validation of the assumptions of the Markov traffic model in cases of real traffic data and the possible applications of our simulation approach,

e.g., modelling the pollution or energy consumption in a city due to multi-modal traffic with gasoline, diesel, electric and plug-in hybrid vehicles, as well as public transportation.

Supporting information

S1 Appendix.
(PDF)

Acknowledgments

The authors would like to thank all actual and former members of the smart city group of the University of Debrecen. Special thanks to Louis Mattia for a close reading of the manuscript. We are especially grateful to all of the participants of the OOCWC competitions and the students of the BSc courses of "High Level Programming Languages" at the University of Debrecen.

Author Contributions

Conceptualization: Márton Ispány.

Data curation: Péter Jeszenszky, Roland Major, Fanny Monori.

Formal analysis: Péter Jeszenszky, Márton Ispány.

Funding acquisition: Renátó Besenczi, Norbert Bátfai, Márton Ispány.

Investigation: Renátó Besenczi, Márton Ispány.

Methodology: Renátó Besenczi, Márton Ispány.

Project administration: Márton Ispány.

Software: Renátó Besenczi, Norbert Bátfai.

Supervision: Márton Ispány.

Validation: Renátó Besenczi, Márton Ispány.

Visualization: Péter Jeszenszky.

Writing – original draft: Renátó Besenczi, Péter Jeszenszky, Fanny Monori, Márton Ispány.

Writing – review & editing: Renátó Besenczi, Márton Ispány.

References

1. Albino V, Berardi U, Dangelico RM. Smart Cities: Definitions, Dimensions, Performance, and Initiatives. Journal of Urban Technology. 2015; 22(1):3–21. https://doi.org/10.1080/10630732.2014.942092
2. Ismagilova E, Hughes L, Dwivedi YK, Raman KR. Smart cities: Advances in research—An information systems perspective. International Journal of Information Management. 2019; 47:88–100. https://doi.org/10.1016/j.ijinfomgt.2019.01.004
3. Zheng Y, Capra L, Wolfson O, Yang H. Urban Computing: Concepts, Methodologies, and Applications. ACM Transactions on Intelligent Systems and Technology. 2014; 5(3):38:1–38:55. https://doi.org/10.1145/2629592
4. Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M. Internet of Things for Smart Cities. IEEE Internet of Things Journal. 2014; 1(1):22–32. https://doi.org/10.1109/JIOT.2014.2306328
5. World Urbanization Prospects: The 2018 Revision (ST/ESA/SER. A/420). United Nations Department of Economic and Social Affairs; 2019.
6. Çolak S, Lima A, González MC. Understanding congested travel in urban areas. Nature communications. 2016; 7(1):1–8. https://doi.org/10.1038/ncomms10793 PMID: 26978719

7. Olmos LE, Çolak S, Shafiei S, Saberi M, González MC. Macroscopic dynamics and the collapse of urban traffic. Proceedings of the National Academy of Sciences. 2018; 115(50):12654–12661. https://doi.org/10.1073/pnas.1800474115 PMID: 30530677

8. Chodrow PS, Al-Awwad Z, Jiang S, González MC. Demand and congestion in multiplex transportation networks. PloS one. 2016; 11(9):e0161738. https://doi.org/10.1371/journal.pone.0161738 PMID: 27657738

9. Chen Y, Ardila-Gomez A, Frame G. Achieving energy savings by intelligent transportation systems investments in the context of smart cities. World Bank; 2016.

10. Xiong Z, Sheng H, Rong W, Cooper D. Intelligent transportation systems for smart cities: A progress review. Science China Information Sciences. 2012; 55:2908–2914. https://doi.org/10.1007/s11432-012-4725-1

11. Menouar H, Guvenc I, Akkaya K, Uluagac AS, Kadri A, Tuncer A. UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges. IEEE Communications Magazine. 2017; 55(3):22–28. https://doi.org/10.1109/MCOM.2017.1600238CM

12. Xu Y, Çolak S, Kara EC, Moura SJ, González MC. Planning for electric vehicle needs by coupling charging profiles with urban mobility. Nature Energy. 2018; 3(6):484–493.

13. Bátfai N, Besenczi R, Mamenyák A, Ispány M. OOCWC: The Robocar World Championship initiative. In: Planck T, editor. Telecommunications (ConTEL), IEEE 13th International Conference on; 2015. p. 1–6.

14. Bátfai N, Besenczi R, Mamenyák A, Ispány M. Traffic simulation based on the Robocar World Championship initiative. Infocommunications Journal. 2015; 7(3):50–58.

15. Besenczi R, Szilágyi M, Bátfai N, Mamenyák A, Oniga I, Ispány M. Using crowdsensed information for traffic simulation in the Robocar World Championship framework. In: Cognitive Infocommunications (CogInfoCom), 6th IEEE International Conference on; 2015. p. 333–337.

16. Besenczi R, Katona T, Szilágyi M. A fork implementation of the Police Edition of the OOCWC system. In: Cognitive Infocommunications (CogInfoCom), 6th IEEE International Conference on; 2015. p. 163–164.

17. Bátfai N, Besenczi R, Ispány M, Jeszenszky P, Major RS, Monori F. Markov modeling and simulation of traffic flow. In: Data Science, Statistics & Visualisation, DSSV 2018; 2018. p. 61. Available from: http://cstat.tuwien.ac.at/filz/BoA.pdf.

18. Nagel K, Schreckenberg M. A cellular automaton model for freeway traffic. Journal de Physique I France. 1992; 2(12):2221–2229. https://doi.org/10.1051/jp1:1992277

19. Horni A, Nagel K, Axhausen KW. The Multi-Agent Transport Simulation MATSim. Ubiquity Press London; 2016.

20. Krajzewicz D, Erdmann J, Behrisch M, Bieker L. Recent Development and Applications of SUMO—Simulation of Urban MObility. International Journal On Advances in Systems and Measurements. 2012; 5(3&4):128–138.

21. Crisostomi E, Kirkland S, Shorten R. A Google-like model of road network dynamics and its application to regulation and control. International Journal of Control. 2011; 84(3):633–651. https://doi.org/10.1080/00207179.2011.568005

22. Faizrahnemoona M, Schlote A, Maggi L, Crisostomi E, Shorten R. A big-data model for multi-modal public transportation with application to macroscopic control and optimisation. International Journal of Control. 2015; 88(11):2354–2368. https://doi.org/10.1080/00207179.2015.1043582

23. Faizrahnemoon M. Real-data modelling of transportation networks. Hamilton Institute, National University of Ireland Maynooth; 2016.

24. Dabrowski C, Hunt F. Using Markov chain and graph theory concepts to analyze behavior in complex distributed systems. U.S. National Institute of Standards and Technology; 2011.

25. Cavers M, Vasudevan K. Spatio-temporal complex Markov Chain (SCMC) model using directed graphs: Earthquake sequencing. Pure and Applied Geophysics. 2015; 172(2):225–241. https://doi.org/10.1007/s00024-014-0850-7

26. Lesne A. Complex Networks: from Graph Theory to Biology. Letters in Mathematical Physics. 2006; 78:235–262. https://doi.org/10.1007/s11005-006-0123-1

27. Lee S, Fambro DB. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transportation Research Record. 1999; 1678(1):179–188. https://doi.org/10.3141/1678-22

28. Stathopoulos A, Karlaftis MG. A multivariate state space approach for urban traffic flow modeling and prediction. Transportation Research Part C: Emerging Technologies. 2003; 11(2):121–135. https://doi.org/10.1016/S0968-090X(03)00004-4

29. Ghosh B, Basu B, O'Mahony M. Multivariate short-term traffic flow forecasting using time-series analysis. IEEE Transactions on Intelligent Transportation Systems. 2009; 10(2):246. https://doi.org/10.1109/TITS.2009.2021448

30. Xue J, Shi Z. Short-time traffic flow prediction based on chaos time series theory. Journal of Transportation Systems Engineering and Information Technology. 2008; 8(5):68–72. https://doi.org/10.1016/S1570-6672(08)60040-9

31. Wang Y, Papageorgiou M. Real-time freeway traffic state estimation based on extended Kalman filter: A general approach. Transportation Research Part B: Methodological. 2005; 39(2):141–167. https://doi.org/10.1016/j.trb.2004.03.003

32. Ngoduy D. Low-rank unscented Kalman filter for freeway traffic estimation problems. Transportation Research Record. 2011; 2260(1):113–122. https://doi.org/10.3141/2260-13

33. Davis GA, Nihan NL. Nonparametric regression and short-term freeway traffic forecasting. Journal of Transportation Engineering. 1991; 117(2):178–188. https://doi.org/10.1061/(ASCE)0733-947X(1991)117:2(178)

34. Smith BL, Williams BM, Oswald RK. Comparison of parametric and nonparametric models for traffic flow forecasting. Transportation Research Part C: Emerging Technologies. 2002; 10(4):303–321. https://doi.org/10.1016/S0968-090X(02)00009-8

35. Turochy RE, Pierce BD. Relating short-term traffic forecasting to current system state using nonparametric regression. In: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems; 2004. p. 239–244.

36. Smith BL, Demetsky MJ. Traffic flow forecasting: Comparison of modeling approaches. Journal of Transportation Engineering. 1997; 123(4):261–266. https://doi.org/10.1061/(ASCE)0733-947X(1997)123:4(261)

37. Messer CJ. Advanced freeway system ramp metering strategies for Texas. Texas Transportation Institute, College Station, TX; 1993.

38. Castro-Neto M, Jeong YS, Jeong MK, Han LD. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Systems with Applications. 2009; 36(3):6164–6173. https://doi.org/10.1016/j.eswa.2008.07.069

39. Nicholson H, Swann CD. The prediction of traffic flow volumes based on spectral analysis. Transportation Research. 1974; 8(6):533–538. https://doi.org/10.1016/0041-1647(74)90030-6

40. Jiang X, Adeli H. Dynamic wavelet neural network model for traffic flow forecasting. Journal of Transportation Engineering. 2005; 131(10):771–779. https://doi.org/10.1061/(ASCE)0733-947X(2005)131:10(771)

41. Xie Y, Zhang Y. A wavelet network model for short-term traffic volume forecasting. Journal of Intelligent Transportation Systems. 2006; 10(3):141–150. https://doi.org/10.1080/15472450600798551

42. Cheng Y, Zhang Y, Hu J, Li L. Mining for similarities in urban traffic flow using wavelets. In: 2007 IEEE Intelligent Transportation Systems Conference; 2007. p. 119–124.

43. Jeong YS, Byon YJ, Castro-Neto MM, Easa SM. Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems. 2013; 14(4):1700–1707. https://doi.org/10.1109/TITS.2013.2267735

44. Chan KY, Dillon TS, Singh J, Chang E. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg-Marquardt algorithm. IEEE Transactions on Intelligent Transportation Systems. 2012; 13(2):644–654. https://doi.org/10.1109/TITS.2011.2174051

45. Park B, Messer CJ, Urbanik T. Short-term freeway traffic volume forecasting using radial basis function neural network. Transportation Research Record. 1998; 1651(1):39–47. https://doi.org/10.3141/1651-06

46. Dia H. An object-oriented neural network approach to short-term traffic forecasting. European Journal of Operational Research. 2001; 131(2):253–261. https://doi.org/10.1016/S0377-2217(00)00125-9

47. Sun S, Zhang C, Yu G. A Bayesian network approach to traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems. 2006; 7(1):124–132. https://doi.org/10.1109/TITS.2006.869623

48. Lv Y, Duan Y, Kang W, Li Z, Wang FY. Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems. 2015; 16(2):865–873.

49. Brameier MF, Banzhaf W. Basic concepts of linear genetic programming. Linear Genetic Programming. 2007; p. 13–34.

50. Iokibe T, Mochizuki N, Kimura T. Traffic prediction method by fuzzy logic. In: Second IEEE International Conference on Fuzzy Systems; 1993. p. 673–678.

51. Li L, Lin WH, Liu H. Type-2 fuzzy logic approach for short-term traffic forecasting. In: IEEE Proceedings-Intelligent Transport Systems. vol. 153. IET; 2006. p. 33–40.

52. Zhang Y, Ye Z. Short-term traffic flow forecasting using fuzzy logic system methods. Journal of Intelligent Transportation Systems. 2008; 12(3):102–112. https://doi.org/10.1080/15472450802262281

53. Necula E. Dynamic traffic flow prediction based on GPS data. In: IEEE 26th International Conference on Tools with Artificial Intelligence; 2014. p. 922–929.

54. Hjort NL, Varin C. ML, PL, QL in Markov chain models. Scandinavian Journal of Statistics. 2008; 35 (1):64–82. https://doi.org/10.1111/j.1467-9469.2007.00559.x

55. Bang-Jensen J, Gutin GZ. Digraphs: Theory, Algorithms and Applications. Berlin Heidelberg New York: Springer Science & Business Media; 2008.

56. Pan B, Zheng Y, Wilkie D, Shahabi C. Crowd sensing of traffic anomalies based on human mobility and social media. In: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM; 2013. p. 344–353.

57. Crovella M, Kolaczyk E. Graph wavelets for spatial traffic analysis. In: IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428). vol. 3; 2003. p. 1848–1857.

58. Wu Y, Zhang X, Bian Y, Cai Z, Lian X, Liao X, et al. Second-order random walk-based proximity measures in graph analysis: Formulations and algorithms. The VLDB Journal. 2018; 27(1):127–152. https://doi.org/10.1007/s00778-017-0490-5

59. Porta S, Crucitti P, Latora V. The Network Analysis of Urban Streets: A Dual Approach. Physica A. 2006; 369:853–866. https://doi.org/10.1016/j.physa.2005.12.063

60. Faizrahnemoona M, Schlote A, Crisostomi E, Shorten R. A Google-like model for public transport. In: International Conference on Connected Vehicles and Expo (ICCVE); 2013. p. 612–613.

61. Horn RA, Johnson CR. Matrix Analysis. Cambridge: Cambridge University Press; 2012.

62. Asmussen S. Applied Probability and Queues. vol. 51 of Applications of Mathematics (New York). New York: Springer-Verlag; 2003.

63. Brémaud P. Markov chains. Gibbs fields, Monte Carlo simulation, and queues. vol. 31 of Texts in Applied Mathematics. New York: Springer-Verlag; 1999.

64. Jarvis JP, Shier DR. Graph-theoretic analysis of finite Markov chains. In: Shier DR, Wallenius KT, editors. Applied mathematical modeling: A multidisciplinary approach. CRC Press; 1996. p. 85–102.

65. Bishop CM. Pattern Recognition and Machine Learning. Information Science and Statistics. New York: Springer-Verlag; 2006.

66. Johnson NL, Kotz S, Balakrishnan N. Discrete Multivariate Distributions. New York: Wiley; 1997.

67. Zhao X, Spall JC. A Markovian framework for modeling dynamic network traffic. In: 2018 Annual American Control Conference (ACC). June 27–29, 2018. Wisconsin Center, Milwaukee, USA; 2018. p. 6616–6621.

68. Lovász L. Random Walks on Graphs: A Survey. In: Miklós D, Sós VT, Szőnyi T, editors. Combinatorics, Paul Erdős is eighty: Papers from the International Conference on Combinatorics. Budapest, Magyarország: János Bolyai Mathematical Society; 1996. p. 353–397.

69. Von Luxburg U. A tutorial on spectral clustering. Statistics and Computing. 2007; 17(4):395–416. https://doi.org/10.1007/s11222-007-9033-z

70. Billingsley P. Statistical Inference for Markov Processes. Chicago: The University of Chicago Press; 1961.

71. Teodorescu I. Maximum likelihood estimation for Markov chains. Arxiv; 2009.

72. Besenczi R, Bátfai N, Jeszenszky P, Major SR, Monori F, Ispány M. Large-scale Analysis and Simulation of Traffic Flow using Markov Models. Arxiv; 2020.

73. Langville AN, Meyer CD. Google's PageRank and Beyond—The Science of Search Engine Rankings. Princeton, NJ: Princeton University Press; 2006.

74. Matsumoto M, Nishimura T. Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator. ACM Transactions on Modeling and Computer Simulation. 1998; 8(1):3–30. https://doi.org/10.1145/272991.272995