

A Dynamic Programming Approach for Road Traffic Estimation

Mattia Laurini, Irene Saccani, Stefano Ardizzoni, Luca Consolini, and
Marco Locatelli*

Abstract

We consider a road network represented by a directed graph. We assume to collect many measurements of traffic flows on all the network arcs, or on a subset of them. We assume that the users are divided into different groups. Each group follows a different path. The flows of all user groups are modeled as a set of independent Poisson processes. Our focus is estimating the paths followed by each user group, and the means of the associated Poisson processes. We present a possible solution based on a Dynamic Programming algorithm. The method relies on the knowledge of high order cumulants. We discuss the theoretical properties of the introduced method. Finally, we present some numerical tests on well-known benchmark networks, using synthetic data.

1 Introduction

Many municipalities are equipped with systems for measuring traffic flows at main roads and intersections. This data collection is allowed by different types of sensors, such as inductive loop detectors, video detection systems, microwave radar sensors, or LIDAR sensors. We address the problem of reconstructing the paths traveled by users based on available flow measurements.

More precisely, we represent a road network as a directed graph and assume that several traffic flow measurements are available, on all or a subset of the network's arcs. We divide the users into distinct groups, each group taking a different route. The flows of all these user groups are modeled through a set of independent Poisson processes. Our primary objective is to determine the routes taken by each user group and estimate the average rates of the corresponding Poisson processes.

*All authors are with the University of Parma, Department of Engineering and Architecture, Parco Area delle Scienze 181/A, 43124 Parma, Italy. E-mails: {mattia.laurini, irene.saccani, stefano.ardizzoni, luca.consolini, marco.locatelli}@unipr.it.

1.1 Literature review

Our problem falls in the broader one of Network Tomography, which was formulated for the first time by Vardi in 1996 (see [19]). The purpose of solving this problem is to estimate the traffic demand of origin–destination (OD) pairs from the knowledge of traffic flows over some links of a network. Since its first appearance, the problem has been widely studied and many variations have been considered (see, e.g., [12] for a comprehensive survey on the topic). The problem of Network Tomography is important in transportation networks, but also in data communication networks. Indeed, a large part of the relevant literature is related to this field. For instance, Network Topology Tomography deals with the problem of reconstructing the layout of a communication network starting from a set of measurement data. Most of Network Tomography formulations assume that a limited amount of measurements is available, and try to infer link characteristics or shared congestion in the middle of the network. For instance, [7] addresses the problem of selecting a minimal subset of paths to be monitored with the aim of inferring the loss rates and latencies of all other paths of the network.

Traffic flows are usually assumed to be independent Poisson random processes; such assumption allows for a simplification of the problem formulation. In [19], Vardi used a second–order moment matching approach to estimate traffic demands, where a solution for the resulting linear matrix equation was obtained using an iterative minimum I–divergence procedure. Work [13] proposed an extension of the second–order moment matching approach of [19] to higher order cumulants matching. The higher order cumulants introduce new useful information on the unavailable distribution of traffic flows, which can be leveraged in estimating the OD demands. By using a sufficient number of empirical cumulants, the linear mapping involved in the cumulant matching approach can achieve full column rank. In an ideal scenario where the cumulant matching equations are consistent, and the cumulants of the link traffic flows are accurately known, the rate vector can be recovered without error. It was demonstrated in [13] that this ideal situation is approachable when a sufficiently large number of realizations of traffic flows is available. In [10], the demands are estimated by using Moment Generating Function (MGF) matching. This is an extension of the moment matching approach to a function that represents moments of all orders. Finally, the Poisson assumption, although common, may constitute an oversimplification in some contexts, like internet communication networks, and can be, hence, relaxed. For instance, work [9] assumes that each OD traffic demand has a mixed Poisson distribution.

1.2 Statement of contribution

The contribution of this work lies in the development of a novel approach which can be applied for estimating traffic demands and roads usage in a network. Specifically, it provides a solution to the OD Tomography problem together with that of identifying the paths with non–zero flow within a network.

In order to achieve our aim, we define a partial order on what can be regarded as the paths set, and define a monotone non–increasing function on it. We derive

some theoretical properties of the partial order and the non-increasing function. Then, exploiting these results, we define a Dynamic Programming (DP) procedure which allows efficiently solving our estimation problem. In this method, the paths set constitutes the state space explored by the DP. To the authors' knowledge, the DP approach on which the method is based is novel in the field of traffic matrix tomography and provides a new and efficient way for solving this problem.

The devised algorithm has been tested on well-known benchmark networks using synthetic data, providing a basis for comparison with existing methods which will be addressed in future works. Ultimately, this work introduces an advancement in traffic matrix tomography presenting an algorithm that combines a technique typical of Computer Science and Control Theory, such as DP, and an important statistical instrument, such as Cumulant Generating Functions (CGFs).

Unlike other approaches, in which only traffic demands are estimated (see, for instance, [1, 2, 17]) or in which the topology of the network is unknown and one wishes to reconstruct it (see, e.g., [3, 6]), in this work, the topology of the network is assumed to be known a priori. Note that such assumption is reasonable since the practical application refers to vehicular traffic networks. Hence, the only quantities to be determined, in addition to traffic demands, are the paths that users follow. The estimation of both these quantities can account for a more detailed analysis of traffic behavior and network usage. It is important to remark that, if paths connecting multiple OD pairs pass through the same set of measured arcs, then we are not able to determine and separate the traffic demand contribution of each of these OD pairs individually but only their aggregate contribution.

This work also relates to [15], which also uses higher-order cumulants and exploits a heuristic procedure in order to reduce the number and order of the computed cumulants, which is critical in terms of numerical estimation. In our work, we also employ higher-order cumulants, however, from a theoretical point of view our method is exact in the sense that it computes deterministically all and only the cumulants needed for obtaining a solution to our problem.

1.3 Paper organization

The paper is organized as follows: in Section 2, we present the estimation problem of this work in a formal algebraic setting, introduce the concept of CGFs, which is key for addressing this problem, and, under the Poisson assumption, show how their formulation can be simplified. In Section 2.2, we discuss in further detail the addressed problems, introducing some auxiliary functions and stating some of their properties. Furthermore, in Algorithm 1 we present the method based on DP for solving the problem of our interest. In Section 3, we show how the problem of inferring paths and estimating demands of a traffic network falls into the class of problems presented in Section 2 and, hence, can be solved through the method introduced in Section 2.2. In Section 4, we test the devised algorithm on the widely used benchmark networks of Sioux Falls and NSFnet and show the obtained results. Finally, in Section 5, we draw the conclusions and future developments of this work.

1.4 Notation

We denote the set of real numbers by $\mathbb{R}$. Given $X \in \mathbb{R}^q$, we denote the transpose of X by X^T . Given $A = (a_{i,j}) \in \mathbb{R}^{\ell \times q}$, for $i \in \{1, \dots, \ell\}$, we denote the i -th row of A by $a_{i,*}$. For $j \in \{1, \dots, q\}$, X_j represents the j -th component of X , while $a_{*,j}$ is the j -th column of A . Symbols $a_{i,j}$ or $(A)_{i,j}$ denotes the (i,j) -th element of A . Let $X, Y \in \mathbb{R}^q$, we denote with $X \odot Y$ the Hadamard product (or the component-wise product) of X and Y , that is, $(\forall i \in \{1, \dots, q\}) (X \odot Y)_i = X_i \cdot Y_i$.

Let $X \in \mathbb{R}^q$ be a vector of random variables, $E[X] = (\lambda_1, \dots, \lambda_q)^T = \lambda$ denotes the expected value (or mean) of X . Such quantity is also known as the first moment of X . In general, we can define moments of higher order as follows, let $k \in \mathbb{N}$, a moment of order k is $E[X_1^{k_1} \dots X_q^{k_q}]$, with $k_1 + \dots + k_q = k$. Moreover, given a random vector $X \in \mathbb{R}^q$, we define $M_X : \mathbb{R}^q \rightarrow \mathbb{R}$, $M_X(t) = E[e^{t^T X}]$ as the *Moment Generating Function* of X , and consequently, $K_X : \mathbb{R}^q \rightarrow \mathbb{R}$, $K_X(t) = \log M_X(t)$ as the *Cumulant Generating Function (CGF)* of X .

A directed graph is a couple $G = (V, E)$, where V is the node set and $E \subseteq V \times V$ is the edge set. We denote the number of nodes and directed edges by $n = |V|$ and $m = |E|$, respectively. Let $s, t \in V$, a path p from s to t is a sequence of adjacent edges without repetitions with starting node s and final node t . Let P be the set of all paths, we denote by P_s^t the subset of P consisting of those paths that connect origin s to destination t . We denote by $\mathcal{D}_P = \{(s_1, t_1), \dots, (s_d, t_d)\} \subseteq V \times V$, with $d \in \mathbb{N}$, the pairs of starting and final nodes appearing in the paths of P . Note that $P = \bigcup_{(s,t) \in \mathcal{D}_P} P_s^t$.

2 Problem formulation

Let $X = (X_1, X_2, \dots, X_q)^T$ be a vector of n real-valued mutually independent random variables with a Poisson distribution. Let $A = (a_{i,j}) \in \{0, 1\}^{\ell \times q}$ be such that the columns of A are all different from each other. Assume that the random vector Y satisfies the linear relation

$$Y = AX. \quad (1)$$

Since the components of random vector X are independent Poisson processes, X is completely defined by its mean vector $E[X] = (\lambda_1, \lambda_2, \dots, \lambda_q)^T = \lambda$. In our application, typically, q is much larger than ℓ . In components, we write $Y = (Y_1, \dots, Y_\ell)^T$. We consider two different problems.

1. Assume that we know A and have a full knowledge of the statistics of Y . We want to find $E[X]$.
2. Assume that we have a full knowledge of the statistics of Y , but we do not know A . We want to find $E[X]$ and A .

2.1 Problem 1: we know A and the statistics of Y

Let $K_Y : \mathbb{R}^\ell \rightarrow \mathbb{R}$ be the CGF of Y . Namely, for $t = (t_1, \dots, t_\ell)^T \in \mathbb{R}^\ell$, set

$$K_Y(t) := \log M_Y(t) = \log E \left[e^{t^T Y} \right] = \log E \left[e^{t^T A X} \right].$$

Note that

$$t^T A X = \sum_{j=1}^q \left(\sum_{i=1}^{\ell} a_{i,j} t_i \right) X_j.$$

Since the components of X are independent, we can write

$$\begin{aligned} K_Y(t) &= \log E \left[e^{t^T A X} \right] = \log E \left[\prod_{j=1}^q e^{\left(\sum_{i=1}^{\ell} a_{i,j} t_i \right) X_j} \right] = \\ &= \sum_{j=1}^q \log E \left[e^{\left(\sum_{i=1}^{\ell} a_{i,j} t_i \right) X_j} \right] = \sum_{j=1}^q C_j \left(\sum_{i=1}^{\ell} a_{i,j} t_i \right), \end{aligned}$$

where $C_j : \mathbb{R}^\ell \rightarrow \mathbb{R}$, with $C_j(t) = \log E[e^{X_j t}]$, is the CGF of X_j . Since the components of X have a Poisson distribution, the CGF of X_j is $C_j(t) = \lambda_j(e^t - 1)$. Hence,

$$K_Y(t) = \sum_{j=1}^q \lambda_j \left(e^{\left(\sum_{i=1}^{\ell} a_{i,j} t_i \right)} - 1 \right),$$

which is linear with respect to λ_j . Recall that $\lambda = (\lambda_i) \in \mathbb{R}^q$. Note that

$$K_Y(t) = [f_1(t), \dots, f_q(t)] \lambda, \quad (2)$$

where

$$f_j(t) = e^{\left(\sum_{i=1}^{\ell} a_{i,j} t_i \right)} - 1.$$

Note that (2) is a functional equality that needs to be satisfied for all $t \in \mathbb{R}^\ell$. In practice, one can pick a large although finite subset of $\mathbb{R}^\ell$ and solve (2) on such set.

It remains to check if we can solve equation (2) with respect to λ . We can do it if and only if functions $f_1, \dots, f_q$ are independent. The following is a direct consequence of a Theorem due to Borel (see [5]).

Proposition 2.1. *Functions $f_1, \dots, f_q$ are independent if and only if all columns of A are different.*

As a direct consequence, we have the following result.

Proposition 2.2. *In model (1), under the assumptions mentioned above, if we know the full statistics of Y and matrix A , it is possible to compute the mean of X if and only if the columns of A are different.*

In particular, if the columns of A are different, equation (2) allows finding the means vector λ . Note that this result is *not new*. For instance, it was already presented in [19], using a different argument.

2.2 Problem 2: we only know the statistics of Y

We denote by $\mathcal{B} = \{0, 1\}^\ell$ the set of logic vectors with ℓ components and set $\mathcal{B}_0 = \mathcal{B} \setminus \{(0, \dots, 0)^T\}$. Note that the cardinality of $\mathcal{B}$ is $|\mathcal{B}| = 2^\ell$ and $|\mathcal{B}_0| = 2^\ell - 1$. We can assign a partial order (see e.g., [8]) on $\mathcal{B}_0$ as follows. Let $v, w \in \mathcal{B}_0$, with $v = (v_1, \dots, v_\ell)^T$, $w = (w_1, \dots, w_\ell)^T$. Then, we set

$$w \leq v \stackrel{\text{def}}{\iff} (\forall i \in \{1, \dots, \ell\}) w_i \leq v_i.$$

In other words, $w \leq v$ if each component of w is lower or equal to the corresponding component of v . Moreover, we write $w < v$, if $w \leq v$ and $w \neq v$.

In principle, since $\mathcal{B}_0$ is finite, one could apply the method presented in Section 2.1, using a matrix A containing all the possible $2^\ell - 1$ choices for its columns (we exclude the column of all zeros) and proceed by estimating only $E[X]$. However, this approach is possible only if ℓ is very small, due to the exponential growth of $|\mathcal{B}|$ with ℓ . To overcome this problem, we present a method, based on DP, that allows finding the columns of A without enumerating all possible elements of $\mathcal{B}_0$.

It is convenient to represent the columns of A as elements of $\mathcal{B}_0$. We denote by $\mathcal{A} = \{a_{*,1}, \dots, a_{*,q}\} \subseteq \mathcal{B}_0$ the set of columns of A , so that, for any $a_{*,j} \in \mathcal{A}$, with $j \in \{1, \dots, q\}$, X_j is the component of random vector X associated with column $a_{*,j}$. Thus, we can rewrite model (1) as

$$Y = \sum_{j \in \{1, \dots, q\}} a_{*,j} X_j.$$

Further, we set function $\rho : \mathcal{B}_0 \rightarrow \mathbb{R}$, such that $\rho(v) = \lambda_j$, if $(\exists j \in \{1, \dots, q\}) v = a_{*,j}$, otherwise $\rho(v) = 0$. In other words, $\rho(v)$ is the mean of the Poisson process associated with column v .

We denote by ∂_i the partial derivative with respect to t_i , and by $\partial_{i_1, \dots, i_p}$ the partial derivative with respect to $t_{i_1}, \dots, t_{i_p}$.

Let $\mathcal{S} = (v_1, \dots, v_r) \in \mathcal{B}_0^r$ be an ordered set of elements of $\mathcal{B}_0$. We define the logic matrix $M_{\mathcal{S}} \in \{0, 1\}^{r \times r}$, such that

$$(M_{\mathcal{S}})_{i,j} = 1 \stackrel{\text{def}}{\iff} v_i \leq v_j. \quad (3)$$

Proposition 2.3. *For any non-empty $\mathcal{S}$, $M_{\mathcal{S}}$ is invertible.*

Proof. Note that $(\forall i \in \{1, \dots, r\}) (M_{\mathcal{S}})_{i,i} = 1$, since for any $v \in \mathcal{S}$, $v \leq v$ (i.e., relation $\leq$ is reflexive). Further, note that if $i \neq j$, at least one between $(M_{\mathcal{S}})_{i,j}$ and $(M_{\mathcal{S}})_{j,i}$ has to be zero (since relation $\leq$ is antisymmetric). Then, matrix $M_{\mathcal{S}}$ is similar to an upper triangular matrix with all ones on the diagonal, hence, it is invertible. $\square$

In particular, note that, if $\mathcal{S} = \mathcal{A}$, matrix $M_{\mathcal{A}}$ is invertible.

Example 2.4. *Let $\ell = 3$ and consider*

$$\mathcal{B}_0 = \left\{ \begin{bmatrix} v_1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} v_2 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} v_3 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} v_4 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} v_5 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} v_6 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} v_7 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}. \quad (4)$$

If we set $\mathcal{S} = \mathcal{B}_0$, we obtain the following matrix

$$M_{\mathcal{B}_0} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

Observe that $M_{\mathcal{B}_0}$ is upper triangular with diagonal elements all equal to 1 and, hence, as stated in Proposition 2.3, is invertible. It is trivial to check that $M_{\mathcal{S}}$ is invertible for any $\mathcal{S} \subseteq \mathcal{B}_0$.

If $v = (v_1, \dots, v_\ell)^T \in \mathcal{B}_0$, then let $i_1 < \dots < i_p \in \{1, \dots, \ell\}$ be such that, for $i \in \{1, \dots, \ell\}$ $v_i = 1 \iff i \in \{i_1, \dots, i_p\}$. In other words, $i_1, \dots, i_p$ are the indices of the components of v equal to 1. Then, we define a function $\phi : \mathcal{B}_0 \rightarrow \mathbb{R}$ as

$$\phi(v) = \partial_{i_1, \dots, i_p} K_Y(t) \Big|_{t=0}.$$

Function ϕ represents the *joint cumulants* of variables $Y_{i_1}, \dots, Y_{i_p}$. The value of $\phi(v)$ corresponds to the sum of the means for all the columns w of A such that $v \leq w$. That is, the following property holds.

Proposition 2.5.

$$\phi(v) = \sum_{w \in \mathcal{A} | v \leq w} \rho(w).$$

Proof.

$$\begin{aligned} \phi(v) &= \partial_{i_1, \dots, i_p} K_Y(t) \Big|_{t=0} = \partial_{i_1, \dots, i_p} \left(e^{t^T A} - 1 \right) \cdot \lambda \Big|_{t=0} = \\ &= \partial_{i_1, \dots, i_p} \left(e^{[t_1 a_{1,*} + \dots + t_q a_{q,*}]} - 1 \right) \Big|_{t=0} \cdot \lambda = \\ &= (a_{i_1,*} \odot a_{i_2,*} \odot \dots \odot a_{i_p,*}) \odot e^{[t_1 a_{1,*} + \dots + t_q a_{q,*}]} \Big|_{t=0} \cdot \lambda = \\ &= (a_{i_1,*} \odot a_{i_2,*} \odot \dots \odot a_{i_p,*}) \cdot \lambda = \sum_{j=1}^q (a_{i_1,j} \dots a_{i_p,j}) \lambda_j = \\ &= \sum_{\substack{j \in \{1, \dots, q\} \\ v \not\leq w}} (a_{i_1,j} \dots a_{i_p,j}) \lambda_j + \sum_{\substack{j \in \{1, \dots, q\} \\ v \leq w}} (a_{i_1,j} \dots a_{i_p,j}) \lambda_j = \\ &= \sum_{\substack{j \in \{1, \dots, q\} \\ v \not\leq w}} 0 \cdot \lambda_j + \sum_{\substack{j \in \{1, \dots, q\} \\ v \leq w}} 1 \cdot \lambda_j = \sum_{v \leq w} \rho(w). \end{aligned}$$

□

Function ϕ is nonnegative and monotone non-increasing, that is, $v \leq w \implies \phi(v) \geq \phi(w)$.

As a consequence

$$\phi(v) = 0 \implies ((\forall w \in \mathcal{B}_0) v \leq w \implies \phi(w) = 0). \quad (6)$$

Note that $\phi(v)$ can be computed by means of a sum of products of the moments of Y (see, for instance, Equation (3.2.7) in [14]) as follows

$$\phi(v) = \sum_{\Pi \in \wp(\{1, \dots, \ell\})} (|\Pi| - 1)! (-1)^{|\Pi| - 1} \prod_{\pi \in \Pi} E \left[\prod_{i \in \pi} Y_i \right]. \quad (7)$$

Definition 2.6. Let $v \in \mathcal{B}_0$. The upper set of v is

$$\uparrow v = \{w \in \mathcal{B}_0 \mid v \leq w\}.$$

Example 2.7. Let us consider $\ell = 3$, $\mathcal{B}_0$ as in (4), and a random vector Y that satisfies model (1) with

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad E[X] = \lambda = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad (8)$$

and for which we know the full statistics of its components, and, in particular, the cumulants, of Y . Note that, for $i \in \{1, 2, 3\}$, $\phi(v_i) = E[Y_i]$. So, we assume to know the following quantities:

$$\begin{bmatrix} \phi(v_1) \\ \phi(v_2) \\ \phi(v_3) \end{bmatrix} = \begin{bmatrix} E[Y_1] \\ E[Y_2] \\ E[Y_3] \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix}, \quad (9)$$

which correspond to the first order cumulants (or the means) of the components of Y . We also know the second order cumulants (which correspond to the variances of the component of Y):

$$\begin{bmatrix} \phi(v_4) \\ \phi(v_5) \\ \phi(v_6) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \quad (10)$$

Finally, by (6), we know that $\phi(v_7) = 0$, since, for instance $v_7 \in \uparrow v_6$. Note that the joint cumulants of Y can be computed by means of (7). Now, starting from v_7 , down to v_1 , we can compute ρ through Proposition 2.5: $\phi(v_7) = \rho(v_7) = 0$. Then, $\phi(v_6) = \rho(v_6) + \rho(v_7) = 0$, from which follows that $\rho(v_6) = 0$. Analogously, $\rho(v_5) = 0$ and $\rho(v_4) = \phi(v_4) = 1$. Consequently $\rho(v_3) = \phi(v_3) = 2$, $\rho(v_1) = \phi(v_1) - \rho(v_4) = 1$, and $\rho(v_2) = \phi(v_2) - \rho(v_4) = 3$.

Definition 2.8. Define function $\psi: \mathcal{B}_0 \rightarrow \mathbb{R}$ as the solution of the following recurrence equation

$$(\forall v \in \mathcal{B}_0) \psi(v) := \phi(v) - \sum_{w \in \uparrow v \setminus v} \psi(w).$$

Note that ψ satisfies the following relation:

$$(\forall v \in \mathcal{B}_0) \phi(v) = \sum_{w \in \uparrow v} \psi(w). \quad (11)$$

Proposition 2.9. $(\forall v \in \mathcal{B}_0) \psi(v) = \rho(v)$. Moreover, if $\psi(v) \neq 0$, then $v \in \mathcal{A}$.

Proof. By Proposition 2.5 and relation (11) we have that

$$(\forall v \in \mathcal{B}_0) \phi(v) = \sum_{w \in \mathcal{A} | v \leq w} \rho(w) = \sum_{w \in \uparrow v} \psi(w). \quad (12)$$

We rewrite (12) as

$$(\forall v \in \mathcal{B}_0) \sum_{w \in \uparrow v} (\psi(w) - \rho(w)) = 0. \quad (13)$$

Note that, for each $v \in \mathcal{B}_0$, (13) can be seen as the product of the row of $M_{\mathcal{B}_0}$, defined as in (3) over $\mathcal{B}_0$, associated to v with column vector $[\psi(v_1) - \rho(v_1), \dots, \psi(v_{2^\ell-1}) - \rho(v_{2^\ell-1})]^T$. We can then rewrite (13) in matrix form as follows:

$$M_{\mathcal{B}_0} \cdot \begin{bmatrix} \psi(v_1) - \rho(v_1) \\ \vdots \\ \psi(v_{2^\ell-1}) - \rho(v_{2^\ell-1}) \end{bmatrix} = 0.$$

By Proposition 2.3, we have that $M_{\mathcal{B}_0}$ is invertible. This implies that $(\forall v \in \mathcal{B}_0) \psi(v) = \rho(v)$. $\square$

So, from Proposition 2.5 and (12), we have that

$$\begin{bmatrix} \psi(v_1) \\ \vdots \\ \psi(v_{2^\ell-1}) \end{bmatrix} = M_{\mathcal{B}_0}^{-1} \cdot \begin{bmatrix} \phi(v_1) \\ \vdots \\ \phi(v_{2^\ell-1}) \end{bmatrix}. \quad (14)$$

However, the computation of $M_{\mathcal{B}_0}$ and, hence, of $M_{\mathcal{B}_0}^{-1}$ is possible only when ℓ is very small.

Example 2.10. Recalling Example 2.7, we can compute the left-hand-side of (14) using $M_{\mathcal{B}_0}^{-1}$, with $M_{\mathcal{B}_0}$ as in (5), and the values of ϕ over $\mathcal{B}_0$ as in Example 2.7 and obtain

$$\begin{bmatrix} \psi(v_1) \\ \psi(v_2) \\ \psi(v_3) \\ \psi(v_4) \\ \psi(v_5) \\ \psi(v_6) \\ \psi(v_7) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

As we will see in Section 2.3, the algorithm based on DP is more efficient at estimating such values avoiding the computation of ϕ over all elements of $\mathcal{B}_0$ and, hence, computing $M_{\mathcal{S}}^{-1}$ with $\mathcal{S}$ in general much smaller than $\mathcal{B}_0$.

Let $v, w \in \mathcal{B}_0$. We say that w is a successor of v if $v < w$ and there are no $z \in \mathcal{B}_0$ such that $v < z < w$. We denote by $\rightarrow v$ the set of successors of v . Note that, since $\leq$ is a partial order, we can have that $(\exists w_1, w_2 \in \mathcal{B}_0) v < w_1 \wedge v < w_2 \wedge w_1 \not\leq w_2 \wedge w_2 \not\leq w_1$.

Proposition 2.9 justifies the procedure presented in Algorithm 1 based on DP for finding $\mathcal{A}$. The algorithm traverses all elements $v \in \mathcal{B}_0$, for which $\phi(v) > 0$. Here, $\mathcal{Q}$ is a FIFO queue, and $\mathcal{V}$ a subset of $\mathcal{B}_0 \times \mathbb{R}$. In what follows, set $\mathcal{B}_0$ is the state space of the DP procedure, and given any state $v \in \mathcal{B}_0$, the admissible state transitions from v are all and only those to states in $\rightarrow v$.

Algorithm 1 DP for estimating $\mathcal{A}$ and $E[X]$

- 1: Let $v = (0, \dots, 0)^T$. Set $\mathcal{Q} = \{v\}$, $\mathcal{V} = \emptyset$.
 - 2: Remove from $\mathcal{Q}$ its first element v .
 - 3: Let $\{w_1, \dots, w_r\} = \rightarrow v$. For each $i \in \{1, \dots, r\}$, if $\phi(w_i) > 0$, add w_i to $\mathcal{Q}$, and add pair $(w_i, \phi(w_i))$ to $\mathcal{V}$.
 - 4: If $\mathcal{Q} \neq \emptyset$, go to 2. Otherwise the algorithm ends.
-

When the algorithm terminates, let $k = |\mathcal{V}|$ be the cardinality of $\mathcal{V} = \{(w_1, \phi(w_1)), \dots, (w_k, \phi(w_k))\}$. Set $\mathcal{S} = (w_1, \dots, w_k)$. Then, the estimated values of x are given by

$$(\psi(w_1), \dots, \psi(w_k))^T = M_{\mathcal{S}}^{-1}(\phi(w_1), \dots, \phi(w_k))^T.$$

Finally, let $\hat{\mathcal{A}} = \{w_i \mid \psi(w_i) > 0\}$.

Proposition 2.11. *The previous algorithm has the following properties.*

- 1) $\mathcal{S} = \{(w, \phi(w)) \mid w \in \mathcal{B}_0, \phi(w) > 0\}$,
- 2) $\hat{\mathcal{A}} = \mathcal{A}$ and $\psi(v) = \rho(v), v \in \mathcal{A}$.

Proof. 1) If $\phi(w) > 0$, there exists a (strictly increasing) sequence $(0, \dots, 0)^T = v_1 < \dots < v_r = w$ such that $(\forall i \in \{2, \dots, r\}) v_i \in \rightarrow v_{i-1}$. Moreover, since $\phi(w) > 0$, being ϕ non-increasing, $(\forall i \in \{1, \dots, r\}) \phi(v_i) > 0$. By induction, note that $(v_1, \phi(v_1)) \in \mathcal{S}$ (step 1). Moreover, for $i \in \{1, \dots, r-1\}$, if $(v_i, \phi(v_i)) \in \mathcal{S}$, $(v_{i+1}, \phi(v_{i+1}))$ is also added to $\mathcal{S}$, since $v_{i+1} \in \rightarrow v_i$ and $\phi(v_{i+1}) > 0$.

- 2) It follows from 1) and Proposition 2.9. □

2.3 Example of Algorithm 1 execution

To clarify how Algorithm 1 works, let us consider Example 2.10, with $\ell = 3$, $\mathcal{B}_0$ defined as in (4), and A and $E[X]$ defined as in (8). We assume that we do not know A and $E[X]$ and want to compute them from the knowledge of the cumulants of Y as given in (9) and (10).

At Step 1, we add v to queue $\mathcal{Q}$ and set $\mathcal{V} = \emptyset$. Then, at Step 2, we remove v_0 from the queue and proceed with Step 2 at computing $\rightarrow v_0 = \{v_1, v_2, v_3\}$ (the successors set of v_0). Then, for each $i \in \{1, 2, 3\}$, if $\phi(v_i) \neq 0$, we add pair $(v_i, \phi(v_i))$ to $\mathcal{V}$ and element v_i to queue $\mathcal{Q}$. In this case, at the end of this step, we have

$\mathcal{Q} = \{v_1, v_2, v_3\}$ and $\mathcal{V} = \{(v_1, 2), (v_2, 4), (v_3, 2)\}$. Since $\mathcal{Q} \neq \emptyset$, we go back to Step 2 and remove v_1 from the queue and compute $\rightarrow v_1 = \{v_4, v_5\}$. At the end of this step $\mathcal{Q} = \{v_2, v_3, v_4\}$ and $\mathcal{V} = \{(v_1, 2), (v_2, 4), (v_3, 2), (v_4, 1)\}$. Note that v_5 has not been added to the queue since $\phi(v_5) = 0$. Again, we remove v_2 from $\mathcal{Q}$, compute $\rightarrow v_2 = \{v_4, v_6\}$ and update $\mathcal{Q} = \{v_3, v_4\}$. Note that at this step, we already have pair $(v_4, 1)$ in $\mathcal{V}$ and no new element is added to $\mathcal{Q}$ or $\mathcal{V}$ since $\phi(v_6) = 0$. Similarly, we remove v_3 from $\mathcal{Q}$, compute $\rightarrow v_3 = \{v_5, v_6\}$ and note that neither $\mathcal{Q}$ or $\mathcal{V}$ need be updated. Finally, we remove v_4 from $\mathcal{Q}$, compute $\rightarrow v_4 = \{v_7\}$ for which we already know that $\phi(v_7) = 0$. Now, since $\mathcal{Q} = \emptyset$, Step 4 terminates the algorithm. We can now compute $k = |\mathcal{V}| = 4$, set $\mathcal{S} = (v_1, v_2, v_3, v_4)$ and obtain $E[X]$ as follows: $[\psi(v_1), \psi(v_2), \psi(v_3), \psi(v_4)]^T = M_{\mathcal{S}}^{-1}[\phi(v_1), \phi(v_2), \phi(v_3), \phi(v_4)]^T \iff$

$$\begin{bmatrix} \psi(v_1) \\ \psi(v_2) \\ \psi(v_3) \\ \psi(v_4) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 4 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix}.$$

Finally, we observe that the vector we just computed equals $E[X]$, matrix $M_{\mathcal{S}}$ is the submatrix of (5) given by its 1st, 2nd, 3rd, and 4th columns and rows, and $(v_1, v_2, v_3, v_4) = A$.

3 Traffic demand and path estimation from flow data

In this section, we consider the problem of estimating traffic demands and user paths from flow measurements. First, we show that this problem can be reduced to the one discussed in Section 2. Let us represent a road network by a directed graph $G = (V, E)$. The edge set $E = \{e_1, \dots, e_m\}$ represents the roads travelled by the users, and the node set $V = \{v_1, \dots, v_n\}$ the intersections and endpoints. Recall that $\mathcal{D}_P$ is the set of OD pairs and P is the set of all paths in G . Let $q = |P|$ be the cardinality of P . We also define the demands vector $X = (X_1, \dots, X_q)^T$. Vector X is structured as follows: a given component X_i is the flow demand associated to a given path $p \in P$. Assuming that the traffic is not congested, we have that, for each OD pair $(s, t) \in \mathcal{D}_P$, all vehicles moving from s to t follow the shortest path in G with respect to travel time. Observe that such assumption is reasonable for road vehicles away from rush hours, or for bicycle networks. Moreover, note that under this condition, for each $(s, t) \in \mathcal{D}_P$, there will be exactly one non-zero component X_j associated with (s, t) , that is, the one associated with path $p \in P_s^t$, with p being the shortest path connecting s to t (assuming the shortest path is unique). For any $(s, t) \in \mathcal{D}_P$, the OD demand of such pair is then given by X_j . Note that in case of congested traffic, for any $(s, t) \in \mathcal{D}_P$ there may be multiple non-zero components of X associated with paths in P_s^t . In such case, the OD demand of such pair is given by the sum of such components (i.e., the traffic demand of a given OD pair is the sum of the traffic flows of all paths connecting that pair). Let $T \in \mathbb{R}^m$ be a vector representing the arcs' total flows. Then, we can write

$$T = MX, \tag{15}$$

where $M = (m_{i,j}) \in \{0,1\}^{m \times q}$ is the edge–path incidence matrix. That is, $m_{i,j} = 1$ if and only if edge e_i belongs to the path associated with component X_j (recalling that, for each $j \in \{1, \dots, q\}$, there exist a path $p \in P$ associated with X_j).

3.1 Problem 1: we observe all flows

In this problem, we assume that we are able to observe traffic flows of all arcs. This means that matrix M in (15) has distinct columns. Indeed, given any path $p = (e_{i_1}, \dots, e_{i_r}) \in P$, there exists exactly one vector of $v \in \mathcal{B}_0$ such that

$$(\forall j \in \{1, \dots, m\}) (v_{i_j} = 1 \iff e_{i_j} \in p). \quad (16)$$

This means that we are in the same hypothesis of (1) and, hence, we can solve the problem of identifying M and X in (15) with the DP procedure presented in Algorithm 1.

Note that Example 2.7, with $\ell = 3$, $\mathcal{B}_0$ as in (4) and A as in (8), can be seen as a problem of form (15) on the following graph $G = (\{a, b, c\}, \{(a, b), (a, c), (b, c)\})$, as depicted in Figure 1. Indeed, $A = (v_1, v_2, v_3, v_4)$, and v_1, v_2, v_3 , and v_4 correspond to paths (a, b) , (b, c) , (a, c) , and $((a, b), (b, c))$, respectively.

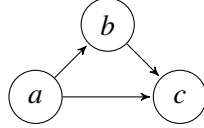


Figure 1: Graph associated with matrix A of (8).

However, assuming that we can observe all traffic flows is not plausible in real–life applications. In fact, when considering real–life traffic networks only a small subset of arc flows is measured, and this motivates the following problem.

3.2 Problem 2: we observe a subset of flows

We now assume that we only measure a subset $\bar{E} = \{e_{i_1}, \dots, e_{i_\ell}\}$ of $\ell \ll m$ arcs of E . Such restriction is realistic in traffic flow measurements in which only a small set of edges flow can be retrieved. To represent this fact, we define a projection matrix $\Pi \in \{0,1\}^{\ell \times m}$, such that $Y = \Pi T$, with $Y \in \mathbb{R}^\ell$ representing the measured flows. Moreover, by setting $A = \Pi M \in \{0,1\}^{\ell \times q}$, we obtain a linear relation similar to (1). However, note that after applying projection Π to M , property (16) does not hold in general. Hence, the hypothesis of (1), which requires that the columns of A are all different, is not guaranteed. More formally, $|\mathcal{A}| = \bar{q} < q$, where q is the number of columns of A and $\bar{q}$ is the number of distinct columns of A . Let us now see how we can reformulate this problem as (1). Let us define relation $\equiv$ in the following way: for any $p, \tilde{p} \in P$, let $v, \tilde{v} \in \mathcal{B}_0$ satisfy (16) for p and $\tilde{p}$, respectively; then

$$p \equiv \tilde{p} \stackrel{\text{def}}{\iff} \Pi(v) = \Pi(\tilde{v}),$$

that is, p and $\tilde{p}$ pass through the same set of arcs in $\bar{E}$. It is easy to verify that $\equiv$ is an equivalence relation (i.e., it is reflexive, symmetric, and transitive). This means

that we can consider the quotient $\bar{P} = P / \equiv$ of P with respect to $\equiv$. An element $\bar{p} \in \bar{P}$ represents all paths in P that pass through the same edges in $\bar{E}$. Since there is a one-to-one map between P and X , we can consider the quotient $\bar{X} = X / \equiv$, in which each component of $\bar{X}$ corresponds to one and only one element of $\bar{P}$. Let $\bar{q} = |\bar{P}|$ and let $\bar{A} \in \{0, 1\}^{\ell \times \bar{q}}$ be such that its column set $\bar{\mathcal{A}}$ coincides with $\mathcal{A}$ (i.e., the columns of $\bar{A}$ are all unique and correspond to those in $\mathcal{A}$). Note that $\bar{X}$ still satisfies the hypothesis of being a random vector of independent Poisson processes. Indeed, each of its components is the sum of the random variables belonging to the same equivalence class and, under the Poisson assumption, this is still an independent Poisson process with mean the sum of the means of the processes of the equivalence class (see, for instance, Theorem 1.1 in Chapter 5 of [16]). Then, we can now consider the following linear relation

$$Y = \bar{A}\bar{X},$$

which satisfies the hypothesis of (1). In this way we can exploit the method based on DP presented in Algorithm 1 for estimating traffic demands and the paths employed by the users of the network. However, we need to better clarify what traffic demands correspond to in this context. Note that when only $\ell < m$ arc flows are observed, one cannot retrieve the demand associated with each and every path of the network, but one can only obtain aggregated demands. More specifically, we can only estimate the demands associated with each element of quotient $\bar{P}$. So we cannot obtain the single contribution of each path but only the aggregated contribution of its equivalence classes.

4 Simulation results on synthetic data

We implemented our algorithm in Matlab. For our tests, we used a computer with an Intel(R) Core(TM) i7-4510U CPU @ 2.60 GHz processor with 16 GB of RAM. We tested Algorithm 1 on two graphs. The first one is not a vehicular network, but a computer networking infrastructure, known as NSFnet [4]. The second one is a well-known traffic network, representing a part of the main roads of Sioux Falls, South Dakota, USA. For both networks, we assumed to know the travel times for each arc and precomputed the shortest paths for each OD pair. Then, for each test, we generated a random vector, representing the means of the independent Poisson processes X_j in (15).

Since these tests are based on simulated data, we know exactly the means of the Poisson processes and the corresponding paths. Therefore, we can directly compare the estimated demands, obtained with our method, with the actual demands. Note that this is not the case for real-life scenarios, where flow measurements are affected by errors, and the true traffic demands and the corresponding paths are often unknown. In this section, we conduct tests on the problem outlined in Section 3.1, where we assume to measure the flow for all the arcs. For each test, we randomly generate a set of flow measurements, according to model (15). We considered four different total numbers of measurements, logarithmically distributed between 1000 and $3 \cdot 10^6$. In each test, we assumed to have only k OD pairs with a

flow demand higher than 0. Moreover, we assumed that all traffic flow associated with each OD pair follows the shortest path. We generated the flow measures as random numbers from a Poisson distribution, using the MATLAB function *poissrnd*. We computed the error as the difference between the true demands and the estimated ones.

4.1 NSFnet

This network consists of $n = 14$ nodes and $m = 21$ arcs, as shown in Figure 2. Although this graph does not represent a road network, it is widely used in Network Tomography and provides a good example for small-sized tests. Figure 3 shows the



Figure 2: Graph of NSFnet.

error on the total flow (i.e., the sum of all demands of all OD pairs). For NSFNet, the number k of OD pairs with flow demand higher than 0 has 4 values (5, 13, 35, 91), logarithmically spaced. Note that the total number of OD pairs is 91. In Figure 3, we can observe that the error decreases as the number of measurements increases. This is due to the estimation of the joint cumulants $\phi(v)$, which is more precise with a higher number of measurements.

4.2 Sioux Falls

The Sioux Falls traffic network has been obtained from the *Transportation Networks for Research* repository [18]. This network has $n = 24$ nodes and $m = 76$ arcs, as depicted in Figure 4.

The number k of OD pairs has 4 values (5, 14, 37, 100), logarithmically spaced. Figure 5 reports the error divided by the total flow, for every number of measurements, with means indicated in red.

Note that the mean error is higher compared to that on the NSFNet network. This is likely due to the fact that this graph has more arcs, and generally longer

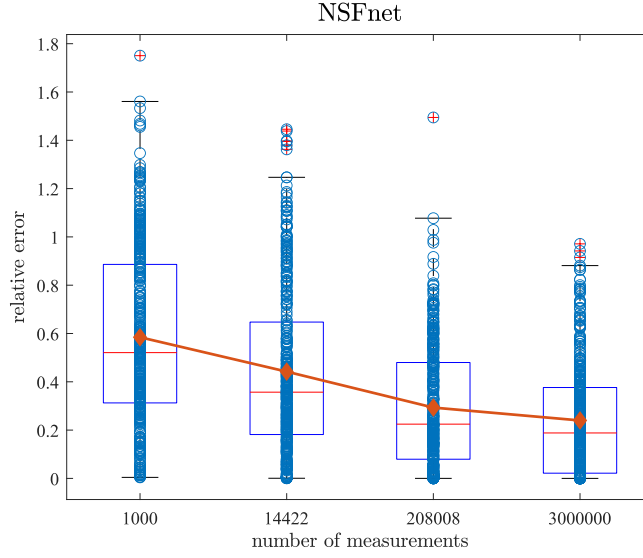


Figure 3: Relative error for 100 tests, on number of measurements. The means for each number of measurements are depicted in red.



Figure 4: Graph of Sioux Falls.

paths, than the NSFNet network. Thus, the execution of Algorithm 1 $\phi(v)$ requires the estimation of higher order joint cumulants, which are more sensitive to outliers, and have a larger mean error.

5 Conclusions

In this paper, we proposed a new method based on DP for both estimating the OD traffic demands and inferring the paths followed by the network users. We

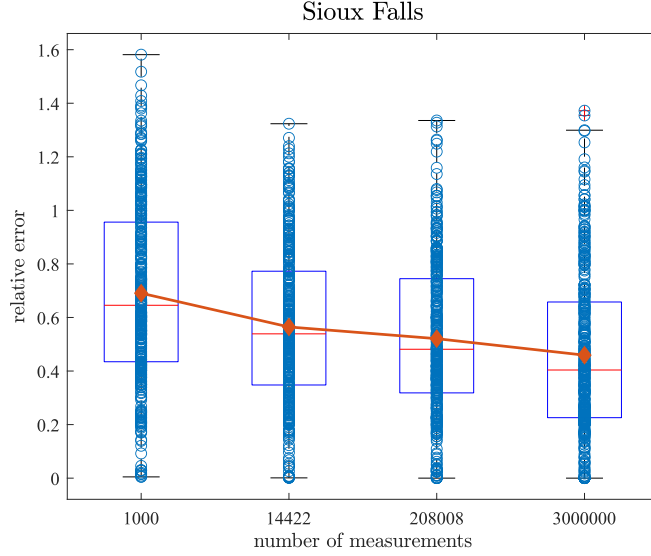


Figure 5: Relative error for 100 tests, on number of measurements. The means for each number of measurements are depicted in red.

described how the newly introduced algorithm exploits the use of higher-order cumulants in order to estimate the desired quantities.

We presented some theoretical results for this method and tested it on synthetic settings. In particular, we applied the algorithm to widely used traffic networks with synthetic data. This allowed us to test the method in a setting in which the exact solution is known. Hence, this allowed us to compare the results obtained with our algorithm with the solutions of the synthetic tests.

Despite the theoretical discussion, in practical applications one needs to take into account some computational aspects. In particular, the computation of cumulants of order higher than three can both be affected by numerical instability and be sensitive to noise and outliers [11]. As a future development, we want to adapt our algorithm in order to take into account this fact by finding a good compromise between exploiting higher order cumulants while, at the same time, avoiding the drawbacks of cumulants of “too high” order, with the aim of making the procedure more robust.

Acknowledgments

Project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5 – Call for tender No. 3277 of 30/12/2021 of Italian Ministry of University and Research funded by the European Union – NextGenerationEU. Award Number: Project code ECS00000033, Concession Decree No. 1052 of 23/06/2022 adopted by the Italian Ministry of University and Research, CUP D93C22000460001, “Ecosystem for Sustainable Transition in Emilia-Romagna” (Ecosister).

References

- [1] M. V. Aerde, H. Rakha, and H. Paramahamsan. Estimation of origin–destination matrices: Relationship between practical and theoretical considerations. *Transportation Research Record*, 1831(03–2296):122–130, 2014.
- [2] S. Bera and K. V. K. Rao. Estimation of origin-destination matrix from traffic counts: the state of the art. *European Transport*, 49:3–23, 2011.
- [3] G. Berkolaiko, N. Duffield, M. Ettehad, and K. Manousakis. Graph reconstruction from path correlation data. *Inverse Problems*, 35(015001), 2019.
- [4] L. H. Bonani. Modeling an optical network operating with hybrid-switching paradigms. *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, 15(4):275–291, December 2016.
- [5] E. Borel. Sur les zéros des fonctions entières. *Acta mathematica*, 20(1):357, 1897.
- [6] R. Cáceres, N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. Loss-based inference of multicast network topology. In *Proceedings of the IEEE Conference on Decision and Control*, February 1999.
- [7] Y. Chen, D. Bindel, H. Song, and R. H. Katz. An algebraic approach to practical and scalable overlay network monitoring. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 55–66, August 2004.
- [8] B. Davey and H. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.
- [9] Y. Ephraim, J. Coblenz, B. L. Mark, and H. Lev-Ari. Mixed poisson traffic rate network tomography. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, 2021.
- [10] Y. Ephraim, J. Coblenz, B. L. Mark, and H. Lev-Ari. Traffic rate network tomography via moment generating function matching. In *2022 56th Annual Conference on Information Sciences and Systems (CISS)*, pages 148–153, March 2022.
- [11] P. Filzmoser, R. Maronna, and M. Werner. Outlier identification in high dimensions. *Computational Statistics & Data Analysis*, 52:1694–1711, 2008.
- [12] T. He, L. Ma, A. Swami, and D. Towsley. *Network Tomography: Identifiability, Measurement Design, and Network State Inference*. Cambridge University Press, 2021.
- [13] H. Lev-Ari, Y. Ephraim, and B. L. Mark. Traffic rate network tomography with higher-order cumulants. *Networks*, 81:220–234, 2023.

- [14] G. Peccati and M. S. Taqqu. *Wiener Chaos: Moments, Cumulants and Diagrams*. Bocconi & Springer Series. Springer Milano, 2011.
- [15] K. D. Smith, S. Jafarpour, A. Swami, and F. Bullo. Topology inference with multivariate cumulants: The möbius inference algorithm. *IEEE/ACM Transactions on networking*, 30(5):2102–2116, 2022.
- [16] H. M. Taylor and S. Karlin. *An Introduction To Stochastic Modeling*. Academic Press, third edition, 1998.
- [17] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):557–573, June 1998.
- [18] Transportation Networks for Research Core Team. Transportation networks for research, <https://github.com/bstabler/transportationnetworks>, February 2024.
- [19] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91(433):365–377, 1996.