HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Networking Laboratory

Ilmari Juva

# Traffic Matrix Estimation

This thesis has been submitted for official examination for the degree of Licentiate of Technology

Espoo, 30.11.2005

Supervisor: Professor Jorma Virtamo

| | |
|---|---|
| Author: | Ilmari Juva |
| Title: | Traffic Matrix Estimation |
| Date: | 30.11.2005 |
| Number of Pages: | 92+3 |
| Department: | Electrical and Communications Engineering |
| Chair: | S-38 Networking Technology |
| Supervisor: | Professor Jorma Virtamo |
| Instructor: | PhD Samuli Aalto |



In a communication network, the traffic has a source where that particular traffic flow is originating from and a destination where it is terminating at. Each origin-destination combination constitutes an origin destination (OD) pair. The knowledge of the amount of traffic of each such OD pair in the network is represented by the *traffic matrix*. The traffic matrix is a required input in many network management and traffic engineering tasks, where in many cases the knowledge on the traffic volumes are assumed to be known. However, in reality, they are seldom readily obtainable in networks, as only the link count measurements and routing information is available. Solving the OD counts from these is an underconstrained problem. Thus, it is not solvable, unless some extra information is brought into the situation.

This thesis gives a comprehensive overview of the estimation methods proposed in the literature. These are divided into a few main groups based on the nature of the extra information each approach uses. The methods based on the gravity model assume that the traffic between two nodes is proportional to the product of the total traffic volumes of the nodes. In the Maximum likelihood methods the sample covariance of link counts is used. In the Bayesian methods there is an assumption about a prior distribution for the estimate. The thesis describes each proposed method and reviews the comparative studies made to evaluate the performance of the methods.

We propose a novel method for traffic matrix estimation: The Quick method based on link covariances, which yields an analytical expression for the estimate and is thus computationally light-weight. The accuracy of the method is compared with that of other methods using second moment estimates by simulation under synthetic traffic scenarios.


Keywords: Traffic matrix estimation, origin-destination traffic, traffic characterization

| | |
|---|---|
| Tekijä: | Ilmari Juva |
| Työn nimi: | Liikennematriisin estimointi |
| Päivämäärä: | 30.11.2005 |
| Sivumäärä: | 92+3 |
| Osasto: | Sähkö- ja Tietoliikennetekniikka |
| Professuuri: | S-38 Tietoverkkotekniikka |
| Valvoja: | Professori Jorma Virtamo |
| Ohjaaja: | FT Samuli Aalto |

Tietoliikenneverkossa kulkevalla liikenteellä on tietty lähtöpiste, josta kyseinen liikennevirta on lähetetty, ja määränpää, johon se on matkalla. Jokainen lähtöpiste ja määränpää muodostaa niin sanotun OD-parin. *Liikennematriisi* sisältää jokaisen tällaisen OD-parin liikennemäärän. Monissa verkon suunnitteluun ja liikenteenhallintaan liittyvissä tehtävissä tieto liikennematriisista on välttämätön. Monesti tämä oletetaan tunnetuksi, mutta todellisuudessa liikennematriisi on harvoin suoraan saatavilla. Yleensä vain linkkien liikennemäärät ja reititystaulut ovat tiedossa. Liikennematriisin ratkaiseminen näillä tiedoilla on alimäärätty tehtävä, eikä siis ratkaistavissa, ellei jotain lisäinformaatiota tilanteesta ole käytettävissä.

Tässä työssä annetaan kattava yleiskuva esitetyistä estimointimenetelmistä. Nämä voidaan jakaa muutamaan pääryhmään sen perusteella, millaista lisäinformaatiota on käytetty. Gravitaatiomalliin perustuvat menetelmät olettavat liikennemäärän kahden pisteen välillä olevan suoraan verrannollinen kyseisten pisteiden kokonaisliikennemääriin. Suurimman uskottavuuden menetelmissä vaadittava lisäinformaatio saadaan linkkien liikennemäärien otoskovarianssimatriisista olettamalla funktionaalinen yhteys liikennemäärän keskiarvon ja varianssin välille. Bayesilaisissa menetelmissä oletetaan, että priori-jakauma liikennemäärille on käytettävissä. Tämä työ kuvailee jokaisen esitetyn menetelmän ja tutkii vertailuja joita on tehty eri menetelmien tarkkuuksista.

Työssä esitetään uusi linkkikovariansseihin perustuva nopea menetelmä liikennematriisin estimointiin, joka antaa analyyttisen ratkaisun estimaatille ja on siten laskennallisesti kevyt. Menetelmän tarkkuutta verrataan muihin vastaaviin menetelmiin simulaatiotutkimuksella.

Avainsanat: Liikennematriisin estimointi, OD-pari liikenne , liikenteen karakterisointi

# Preface

The work for this Licentiate Thesis was carried out in Networking Laboratory of Helsinki University of Technology as part of the FIT and Fancy projects funded by the Academy of Finland, the IRoNet project funded by Tekes and the Euro-NGI special research project.

I would like to thank first and foremost my supervisor Professor Jorma Virtamo.
I would also like to thank Samuli Aalto, Pirkko Kuusela, Sandrine Vaton and Riikka Susitaival for their collaboration in the research leading up to this thesis.

Furthermore, I thank CSC - the Finnish IT Center for Science - for providing access to Funet network and Markus Peuhkuri for his help in handling the data, and the Statistics Research Department at Bell Labs for providing the Lucent data set.

Finally, Thanks to all Netlab employees for the great atmosphere.

Espoo, 30.11.2005,

Ilmari Juva

# Contents

# Acronyms

- **BGP** Border Gateway Protocol
- **DCM** Discrete Choice Model
- **EM** Expectation Maximization
- **IID** Independently and Identically Distributed
- **IS-IS** Intermediate System - Intermediate System
- **IP** Internet Protocol
- **LININPOS** Linear Inverse Positive
- **LP** Linear Programming
- **MCMC** Markov Chain Monte Carlo
- **MLE** Maximum Likelihood Estimate
- **MPLS** Multi Protocol Label Switching
- **N-Q** Normal-Quantile
- **OSPF** Open Shortest Path First
- **OD** origin-destination
- **POP** Point of presence
- **PCA** Principal Components Analysis
- **SNMP** Simple Network Management Protocol
- **TM** Traffic Matrix
- **WLSE** Weighted Least Square Estimation

# Chapter 1

# Introduction to Traffic Matrix Estimation

*In this chapter we give a short introduction to the area of traffic matrix estimation, background of the field and its importance. The problem setting and difficulties involved are introduced and discussed, using a small toy topology as an example. Finally a short overview of the different estimation approaches is given, leaving a detailed review for chapter 3.*

## 1.1 Background

In a communication network, the traffic that transits through the network has a source where that particular traffic flow is originating from and a destination where it is terminating at. The knowledge of the amount of traffic in the network is represented by the *traffic matrix*. Its elements give the volume of traffic between each of the origin-destination (OD) pairs in the network. The traffic matrix is a required input for the operator in many network management tasks. Such tasks include for instance routing, traffic engineering problems such as balancing the traffic load in the network as evenly as possible for all links, as well as network capacity dimensioning. In many cases the knowledge on the underlying traffic volumes are assumed to be known. However, in reality, they are seldom readily available in networks.

It is widely recognized that accurate traffic matrices representing the traffic demands in the network are crucial for traffic engineering, but only recently have there been

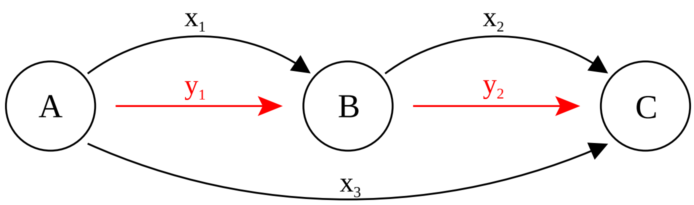


Figure 1.1: A simple two link, three OD-pair topology.

proposals in the literature for methods to obtain such matrices. In the following sections the problem of traffic matrix inference is formulated and we give an overview of some of the different approaches proposed to estimate the traffic matrix.

## 1.2 Introduction

Each traffic flow in a network originates from some origin, and terminates at some destination. These may be links, routers or so called points of presence (POP), depending on the situation, but in the sequel we will refer to these as nodes. Each origin (or source) node $s$ and destination node $d$ constitute an OD pair. The traffic between the origin and destination of an OD pair is denoted by $x_{sd}$, which is the element $(s, d)$ of the traffic matrix $\boldsymbol{x}$. For the computational purposes, the traffic matrix is always written as an $n$-vector $\boldsymbol{x}$, where $n$ is the number of non-zero OD pairs. We refer to the $i$th OD pair by $x_i$. The vector contains all the nonzero elements of the matrix, as zero elements are left out. Let us denote the unknown traffic matrix by stochastic variable $\boldsymbol{X}$ and $\boldsymbol{x}$ stands for some value of this variable.

Consider the simple example network of Figure 1.1. In this topology we have three nodes. To further simplify the situation we have here only two links. One from $A$ to $B$ and the other from $B$ to $C$. We do not consider the links in the opposite directions between the same nodes in this example. Each of the three nodes may serve as an origin and as a destination. Thus we have potentially six origin-destination pairs in the network, namely $AB$, $BA$, $AC$, $CA$, $BC$ and $CB$. The traffic matrix in matrix form would thus be

$$\begin{pmatrix} - & AB & AC \\ BA & - & BC \\ CA & CB & - \end{pmatrix}.$$

But since the network had only links from left to right direction in the figure, there is no traffic from right to left in the example, and half of the above elements of the traffic matrix are zero. We leave the zero elements out and write the nonzero elements in vector form to obtain the OD counts vector of dimension $n = 3$ as

$$\boldsymbol{x} = (\, x_{AB} \quad x_{BC} \quad x_{AC} \,) \,,$$

or we can numerate the elements and write

$$\boldsymbol{x} = (\, x_1 \quad x_2 \quad x_3 \,) \,.$$

These are the actual traffic volumes travelling in the network. The traffic matrix, denoted by $\boldsymbol{\lambda}$ is the expected value of $\boldsymbol{x}$.

$$\mathrm{E}[\boldsymbol{x}] = \boldsymbol{\lambda} = (\, \lambda_1 \quad \lambda_2 \quad \lambda_3 \,) \,.$$

The vector $\boldsymbol{\lambda}$ is what we are trying to estimate in traffic matrix estimation, although in many cases also $\boldsymbol{x}$ is estimated.

### 1.2.1 Available Data

Although direct measurement of traffic matrices is possible with tools like Netflow [36], they are typically not available over the whole network, and network wide use of Netflow would be quite expensive. Hence, the information about the OD-pair volumes $\boldsymbol{x}$ is not readily available, but has to be estimated. What we do have available are the measurements of the traffic in each link, and the routing matrix specifying the path each OD pair uses in the network between nodes $s$ and $d$.

The *link counts*, or link loads, give the measured traffic volumes in each link at a given time. They are denoted by the $m$-vector vector $\boldsymbol{y}$. The element $y_j$ of this vector gives the link count on a specific link $j$. When consecutive measurements are used, we denote the $t$th set of measurements by vector $\boldsymbol{y}_t$. The link counts are obtained from the measurement data available by the Simple Network Management Protocol (SNMP) [8]. The attractive feature of SNMP is that it is usually available everywhere in an IP network. However, it has many limitations, such as possible inaccuracy and unreliability as data may be lost in transport. See [7] for discussion of problems in using SNMP for traffic measurements. Despite the problems, SNMP is the only widespread tool to obtain link count data. The SNMP poller requests periodically each router for the amount of traffic received and transmitted from its interfaces. Typical periods vary from a minute to few minutes, with five minutes being the typical value, although, at least in theory, shorter periods are also possible.

The *Routing matrix* $\boldsymbol{A}$ is of dimensions $m \times n$ and is usually assumed to be known and fixed in traffic matrix estimation problems. Element $A_{j,i}$ of the routing matrix is $1$ if OD pair $i$ uses link $j$, and $0$ otherwise. The routing matrix is obtained from BGP configurations and through OSPF and IS-IS link weight information gathered from the routers.

Consider again the example network of Figure 1.1. Here we have two links. We denote the link from node $A$ to node $B$ as link $1$ and the link from node $B$ to node $C$ as link 2. Thus we have the link count vector of dimension $m = 2$ as

$$\boldsymbol{y} = \begin{pmatrix} y_1 & y_2 \end{pmatrix}.$$

The routing matrix $\boldsymbol{A}$ indicates which links are used by each OD pair. Thus it needs to have dimension $m \times n$, so that each OD pairs and link is covered.

$$\boldsymbol{A} = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \end{pmatrix}.$$

Each column of the routing matrix represents an OD pair, and each row represents a link. For instance, since OD pair $2$ uses link $2$, the element $A_{2,2} = 1$. Looking at the Figure 1.1, we can see that in this case the routing matrix is

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Reading the routing matrix $\boldsymbol{A}$ row by row would thus tell us that link $1$ is used by the first and third OD pair, and link $2$ by the second and third. By reading the routing matrix column by column we can similarly find out which links does a specific OD pair use. For instance, since $A(1,1) = 1$, we know that OD pair $1$ uses link $1$. It does not use link $2$, since $A(2,1) = 0$.

### 1.2.2 The Link Count Relation

Should we know the traffic between OD pairs and the routing matrix, the link counts could easily be calculated. If we again look at our example topology, it is easy to see that link count of the first link comprises of traffic $x_1$ and $x_3$ and the link count of the second link is comprised of $x_2$ and $x_3$.

$$\begin{aligned} y_1 &= x_1 + x_3 \\ y_2 &= x_2 + x_3. \end{aligned} \tag{1.1}$$

Using the routing matrix we can write this in vector form as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1.2}$$

which we call the link count equation.

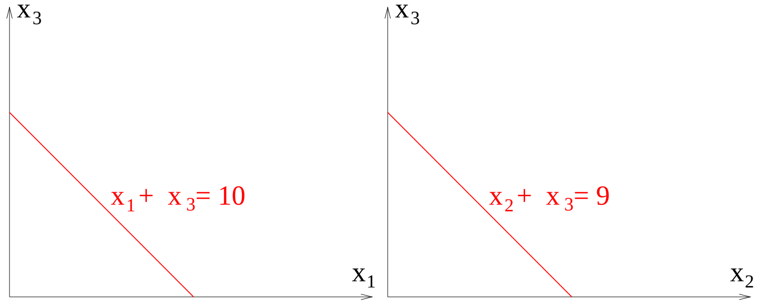


Figure 1.2: In our two link example, there are two link constraints to consider

To understand the meaning of the link count equation we turn back to our example. The two observed link loads $y_i$ each give some indication of the values of the unknown OD loads $x_i$, as formulated in equations (1.1). Suppose the observed link load on link one is $y_1 = 10$ and on link two $y_2 = 9$. These constraints are shown in graphical presentation in Figure 1.2. We now know that the sum of the traffic of OD pairs $1$ and $3$ has to be equal to $10$, that is it lies on the line representing the link count constraint. Equivalently, we get a constraint from the link count measurement on link $2$.

In order to be consistent with the link counts the traffic matrix estimate has to satisfy both of these constraints simultaneously, that is satisfy equation (1.2). The link count constraints in our example are shown in Figure 1.3. As we had three OD pairs, we can present the situation in three dimensional space. Each constraint that was a line in the two-dimensional presentation of Figure 1.2, forms now a plane in the three-dimensional space. The condition $x1 + x3 = 10$ does not depend on $x_2$ and is therefore satisfied for any value of $x_2$. Thus the line shown in the two-dimensional representation is just the intersection of the constraint plane and the $(x_1, x_3)$ co-ordinate plane. In three-dimensional representation the constraint plane continues to the direction of the $x_2$ co-ordinate axis, as depicted in Figure 1.3. Similarly the second constraint is plane, and the space where both conditions hold is the intersection of these two planes, defined by the link load equations, and depicted in the figure by the line labelled $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

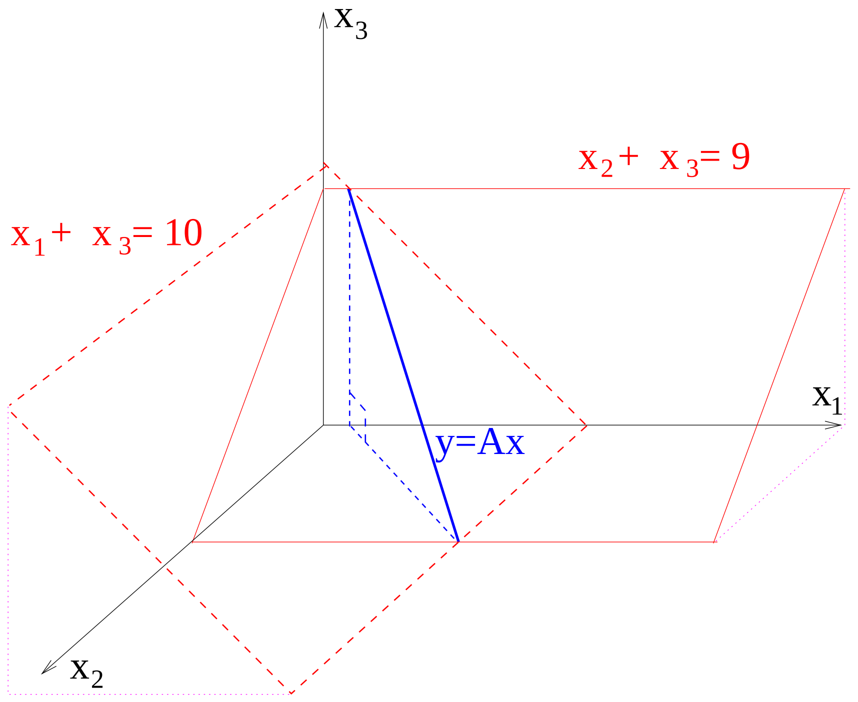


Figure 1.3: In our two link example, the link constraints form a line in three dimensional space

The link count equation 1.2 will hold for any so-called snapshot of the network. In any particular moment of time the link loads are deterministically obtained from the OD loads by this equation. If we have several measurements available we can use the sample average of $\boldsymbol{y}$ as the expected value of the link loads and obtain an equation including the traffic matrix $\boldsymbol{\lambda}$.

$$\mathrm{E}[\boldsymbol{y}] = \boldsymbol{A}\mathrm{E}[\boldsymbol{x}], \tag{1.3}$$

$$\overline{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{\lambda}. \tag{1.4}$$

### 1.2.3 The Traffic Matrix Estimation Problem

The problem setting of traffic matrix estimation can be divided to two different scenarios, depending on the amount of measurements available. Therefore, there is some ambiguity in the literature whether the term *traffic matrix* refers to OD counts $\boldsymbol{x}$ or their expected value $\boldsymbol{\lambda}$.

If only a single measurement snapshot of the link counts is available, the goal of the estimation is to infer the OD counts $\boldsymbol{x}$ from the link counts $\boldsymbol{y}$. On the other hand, if there is a time-series of several link count measurements $\boldsymbol{y}_t$ $(t = 1, \ldots, T)$ available the problem usually is to infer the expected value $\boldsymbol{\lambda}$ of the OD counts, although in some cases the goal is to infer also the link count time series $\boldsymbol{x}_t$ $(t = 1, \ldots, T)$.

**Problem 1 (Snapshot problem)** Given a single set of link counts $\boldsymbol{y}$ and routing matrix $\boldsymbol{A}$, find the traffic matrix $\boldsymbol{x}$ such that conditions $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ are satisfied.

**Problem 2 (Time-series problem)** Given independently and identically distributed link counts $\boldsymbol{y}_t$ $(t = 1, \ldots, T)$ and routing matrix $\boldsymbol{A}$, find the traffic matrix $\boldsymbol{\lambda}$ such that conditions $\overline{\boldsymbol{y}}_t = \boldsymbol{A}\boldsymbol{\lambda}$ are realized as close as possible.

Since in any realistic network there are many more OD pairs than links, the problem of solving $\boldsymbol{x}$ or $\boldsymbol{\lambda}$ from $\boldsymbol{A}$ and $\boldsymbol{y}$ is strongly underdetermined and thus ill-posed. This means that accurate explicit solutions cannot be found, as there are infinite number of solutions that satisfy equation (1.2).

For example, consider a snapshot problem with the measurements

$$y_1 = 10 \qquad y_2 = 9.$$

One possible solutions would be

$$x_1 = 5 \qquad x_2 = 4 \qquad x_3 = 5,$$

another one is

$$x_1 = 4 \qquad x_2 = 3 \qquad x_3 = 6.$$

In fact, for any $a \in [0, 9]$

$$x_1 = 10 - a \qquad x_2 = 9 - a \qquad x_3 = a$$

is a solution that satisfies the link count equation. So even for this small toy example there are $10$ integer solutions, and infinite number of real value solutions.

To overcome this ill-posed nature of the problem, some type of additional information has to be brought in before the problem can be solved. This might be assumptions about the traffic distribution, additional measurements or some prior knowledge about the traffic matrix. For snapshot problems prior the latter is typically used, while time-series problems allow for various approaches.

In Bayesian inference, the estimation depends on the knowledge of a prior distribution for the traffic matrix. The conditional probability distribution for the traffic matrix is then computed given the prior distribution and observed link loads. This method usually employs Markov-chain Monte Carlo simulation for computing the posterior probability. The drawback is that the method is heavily dependent on the accuracy of the prior.

Techniques based on the gravity model make the assumption that source and destination are independent in the sense that if a certain percentage of all traffic goes to destination $d$, the percentage is the same for any given source node $s$.

Maximum likelihood methods assume that there is a relation between the mean and variance of the OD pair counts. This approach therefore requires a time-series of measurements to obtain a sample variance for the link counts. The variance is then used through the relation along with the link counts in a maximum likelihood framework to calculate an estimate for the traffic matrix. The fact that we can gain knowledge about the mean from the sample variance makes the problem identifiable.

### 1.2.4 Contribution of the Thesis and Related Publications

In chapter 2 we analyze measurements from various networks in order to validate or invalidate the key assumptions involved in traffic matrix estimation. Most of the chapter is based on our analysis of novel traffic measurements from the Finnish University network, Funet. These results were first published in [1] and [2].

The main contribution of the Thesis is the Quick estimation method based on link count covariances, which is presented in Chapter 5 and is based on our work in [3]. It proposes an approach to the time-series problem, usually solved by maximum likelihood estimation, where a closed from solution is obtained for the estimate through analytical calculations. While this is not as accurate as maximum likelihood estimate, it is significantly lighter computationally.

### 1.2.5 Structure of This Thesis

In chapter 2 we analyze measurements from real networks to test various common assumptions that are made about the nature of the traffic, in order to simplify the traffic matrix estimation problem. In chapter 3 we give a comprehensive review of the estimation methods proposed in the literature thus far, while chapter 4 reviews comparisons between different methods found in literature currently. In chapter 5 we propose an estimation method of our own, namely the Quick Method based on the link count covariances. Finally, a conclusion of the Thesis is given in chapter 6.

# Chapter 2

# Common Assumptions and Their Validity in Traffic Matrix Estimation

*Various assumptions are commonly made in order to solve the traffic matrix estimation problem. For statistical approaches it is integral to assume a distribution that the unknown traffic volumes are following. Assumptions about static routing and independence between measurements and OD pairs on the other hand are not absolutely necessary, but are made to greatly simplify the estimation problem. The mean-variance relationship allows the use of likelihood methods. In this chapter we explore the validity of these assumptions by studying real traffic traces from the Funet network, Lucent local area network and a commercial network. All the results in this chapter, with the exception of section 2.3, are based on the work we presented in [1] and [2] .*

## 2.1 Data Sets Used in This Chapter

In this section we describe the measurement data sets used in the following sections of this chapter. One is from the Finnish University Network, Funet, another from a local network at Lucent [5], and the third from a commercial network.

### 2.1.1 Funet Data

#### Original Data

The Funet data set consists of link counts measured in one second intervals over two periods; First one from 3am June 29th to 2pm July 6th, and the second one from 10 am August 3rd to 12 am August 10th of 2004 local time. We denote this original measurement data by $x = (x_t\,; t = 1, 2, \ldots, T)$, where $x_t$ refers to the measured link bit count at time $t$ seconds. The link counts are shown in Figure 2.1 for a period of 7 days. A strong diurnal pattern is clearly visible from the figure, as the traffic volume is almost twice as large during the day than during night. Different days look very much the same.

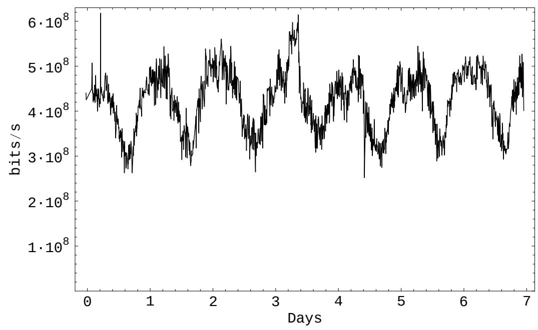


Figure 2.1: Trace of the Funet data set aggregate traffic over one week.

#### Traffic Components

For each time scale $\Delta$ investigated, ranging from $1$ to $300$ seconds, we created the corresponding time series of link counts $x^\Delta = (x_n^\Delta\,; n = 1, 2, \ldots, T/\Delta)$ by defining

$$x_n^\Delta = \frac{1}{\Delta} \sum_{t=n\Delta+1}^{(n+1)\Delta} x_t.$$

As the diurnal variation is much larger than stochastic variation, we separate the traffic into components

$$x_n^\Delta = m_n^\Delta + s_n^\Delta z_n^\Delta,$$

where $m_n^\Delta$ refers to the moving sample-average, $s_n^\Delta$ to the moving sample-standard-deviation. The remaining component $z_n^\Delta$ is then the component of stochastic variation, the sample standardized residual. The averaging period to calculate the sample

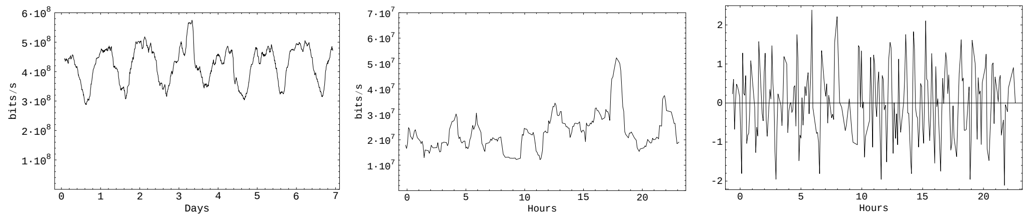


Figure 2.2: From left: sample mean, sample standard deviation and the sample-standardized residual of the Funet trace with $\Delta = 300$ s.

mean and standard deviation was chosen to be one hour. Thus,

$$m_n^\Delta = \frac{1}{3600/\Delta + 1} \sum_{k=n-1800/\Delta}^{n+1800/\Delta} x_k^\Delta$$

and

$$s_n^\Delta = \sqrt{\frac{1}{3600/\Delta + 1} \sum_{k=n-1800/\Delta}^{n+1800/\Delta} (x_k^\Delta - m_k^\Delta)^2}$$

The moving sample-average, $m_n^\Delta$, is depicted as a function of time in left side of Figure 2.2 for the first 7 days. The moving sample-standard-deviation $s_n^\Delta$is depicted for one day period in the middle of the same Figure, while the the standardized random fluctuation $z_n^\Delta$, is shown for the same time period in the right side of Figure 2.2. We will concentrate on this component in further analysis of the traffic characteristics.

### Virtual OD Pairs

For the purposes of sections 2.2.3 and 2.4 the traffic was divided into OD pairs by classifying in terms of IP addresses. We used a 22 bit network mask, meaning that each origin subnetwork could have 10 bits for host part or about 1000 IP addresses. This corresponds to a middle size company. As the resolution here is quite high, there are $2^{22}$ or over four million origin networks. By selecting the $100$ largest networks, we obtain a traffic matrix with $10000$ origin-destination pairs. In practice, only $844$ of these OD pairs had traffic. This is depicted visually in Figure 2.3, which demonstrates the traffic matrix, with black boxes denoting active OD pairs.

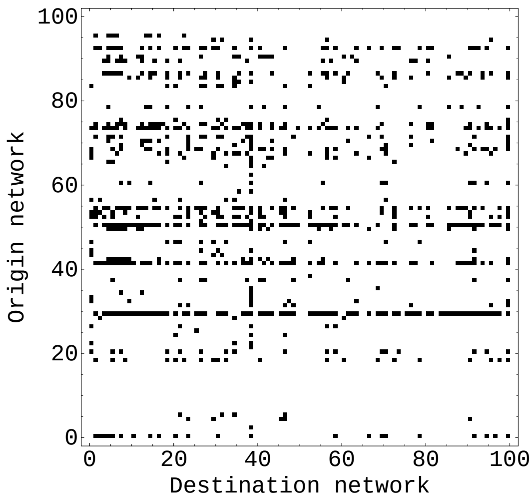


Figure 2.3: Traffic matrix. Black indicates active OD pair, white inactive.

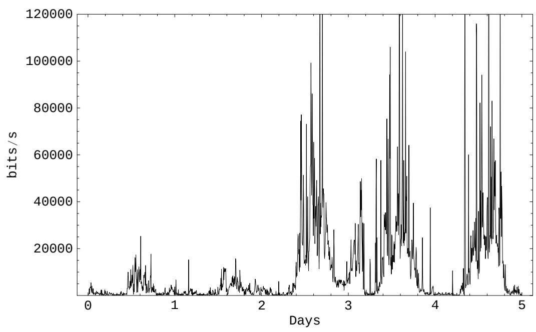


Figure 2.4: Typical OD-pair trace for Lucent data

## 2.1.2 Lucent Data

The Lucent data set contains data from two star topology local area networks, which have $12$ and $5$ nodes, respectively, connected to the routers with one-way links. The measurements consist of OD pair counts over $5$ minute periods, implying that $\Delta = 300$ s for this data. In Figure 2.4 a typical example of an OD-pair trace is shown. The trace is over five days, meaning $1440$ values of five minute aggregates. In some of the OD-pairs the traffic volumes are much smaller over the first two days indicating different traffic rates in the weekends as compared to working days. Another obvious difference to the Funet data is the variability of the traffic. In the Lucent network, traffic is much more variable, and also values close to zero are common for five minute aggregated, as opposed to the Funet link, where even at

$\Delta = 1$ s there were no measurements with zero or close to zero values. This is due to the fact that the Funet link has much more aggregated traffic compared to the local area traffic of Lucent.

### 2.1.3 Commercial ISP Data

The commercial ISP network data set includes one week of link count measurements from a network of $160$ links and $50$ nodes. Each measurement is a $10$ minute aggregate, so altogether there are $1008$ measurements from each link. The OD pair counts are not available, and the measurement period is rather large for the purpose of studying the link count behavior. However, routing tables are also available for each measurement period, and are not the same for all of them. This gives an opportunity to study the behavior of the routing table in a real network.

## 2.2 Gaussian iid Model

### 2.2.1 Testing the Gaussian Assumption

In this section we test whether a Gaussian assumption is valid for the data sets used. This is a typical assumption that simplifies many modelling situations. We concentrate on the stochastic component $z_n^{\Delta}$ as defined in Section 2.1, and study measurements of one second interval, as well as the five minute aggregates.

In Figure 2.5 the histograms of the Funet data are shown, comparing them against the density function of the normal distribution. For the one second time scale the Gaussian density function follows the data nicely. For the five minute aggregates the curve does not follow the histogram as closely, but there is still a reasonably good fit.

A good way to evaluate the appropriateness of the Gaussian assumption is the Normal-quantile (N-Q) plot. The original sample vector $\boldsymbol{x}$ is ordered from the smallest to the largest and plotted against vector $\boldsymbol{a}$, which is defined as

$$a_i = \Phi^{-1}(\frac{i}{n+1}) \qquad i = 1, \ldots, n,$$

where $\Phi$ is the cumulative distribution function of the normal distribution. The values given for $a$ are uniformly distributed between $0$ and $1$, so that the vector $a$

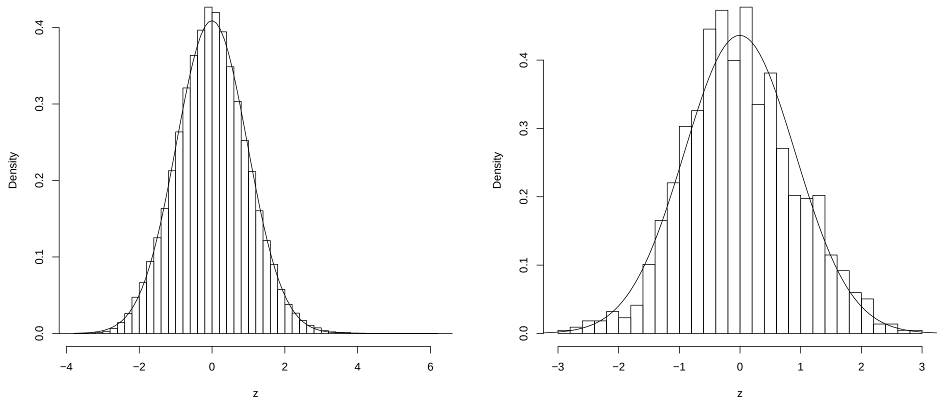


Figure 2.5: Histograms of Funet data with $\Delta = 1\ s$ (left) and $\Delta = 300\ s$ (right) against the normal distribution density function.

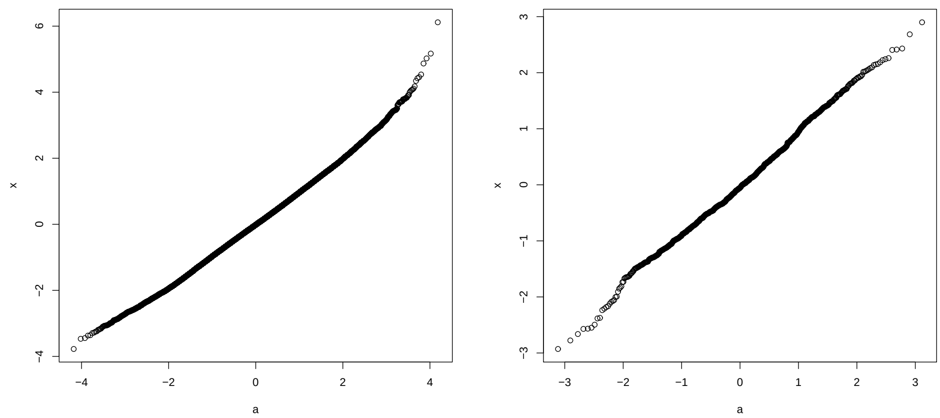


Figure 2.6: N-Q plots comparing the Funet-data with $\Delta = 1\ s$ (left) and $\Delta = 300\ s$ (right) against the normal distribution.

contains the normal quantiles, having values from approximately $-3$ to $3$. If the considered data follows the normal distribution, the plot should be linear. Goodness of fit with respect to this can be calculated by the linear correlation coefficient $r$, and the value $r^2$ is used as a measure of the fit.

$$r(x,a) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(a_i - \overline{a})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(a_i - \overline{a})^2}}.$$

The N-Q plots shown in Figure 2.6 confirm the strong Gaussianity observed in the histograms. The $r^2$-values are $0.999$ and $0.996$ for the one second measurements and the five minutes aggregates respectively. Thus we can conclude that the normal approximation seems to be reasonably good, or at least cannot be rejected based on this.

In [5] the N-Q plot for the Lucent data set is given. The distribution of that data set has heavier tails than the normal distribution and also high peaks around the mean, which causes visible concavity for the N-Q plot. This seems to be the case for most OD pairs in the data set. Cao et al. conclude that the fit is still sufficient for the normal-approximation to be used.

### 2.2.2 Independence of Consecutive Measurements

In this section we study the autocorrelations of the data with respect to different aggregation intervals. For the observations truly to be considered IID, there should

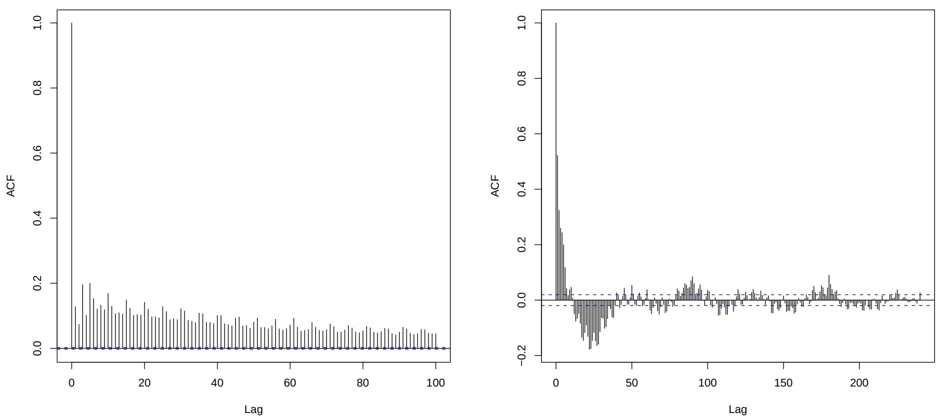


Figure 2.7: Autocorrelations of Funet data. Left: one second measurement interval ($\Delta = 1$ s), right: one minute measurement interval ($\Delta = 60$ s).

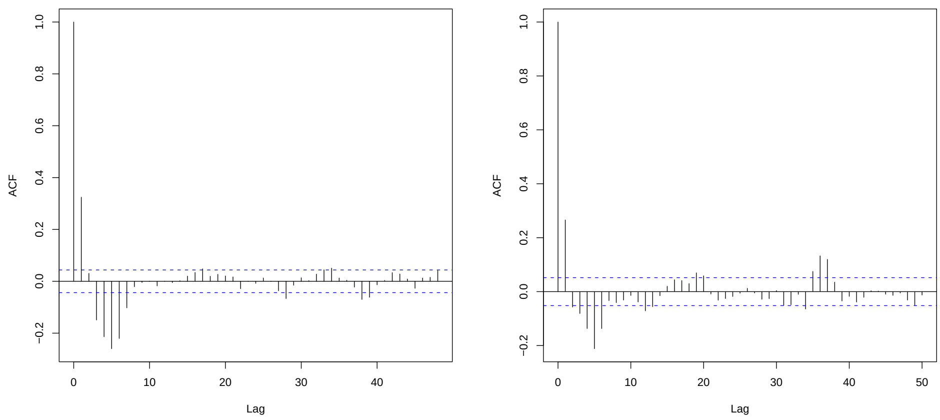


Figure 2.8: Autocorrelation of Funet data with five minute aggregation measurements ($\Delta = 300$ s) and autocorrelation of a typical OD-pair from the Lucent data on the right (bottom).

not be any significant autocorrelations in the stochastic component $z_n^{\Delta}$.

Figure 2.7 shows autocorrelation for the one second and $60$ seconds measurement intervals for one week of the Funet data. Clearly there are positive autocorrelations, indicating dependency between consecutive measurements. In the case of one minute aggregates we notice significant positive values up to a lag of a little over five minutes, and then a set of negative autocorrelation values after that is clearly observable. In Figure 2.8 the autocorrelation of the five minute aggregates of the Funet data as well as the autocorrelation of a typical Lucent data OD-pair are shown. The autocorrelation of the Funet data obviously corresponds nicely to the behavior of the one minute aggregates of the same data set. Comparing Lucent and Funet five minute measurements, a noticeable result is that the autocorrelation function of these two very different data sets seem to be surprisingly similar. In both cases it is not until a lag of more than thirty minutes that there is not any significant autocorrelation.

### 2.2.3 Independence between OD Pairs

In this section we study whether the OD pairs are independent from each other. We concentrate on the dependency between the residual components of the OD pairs, because obviously there would be clear correlation between OD pairs with diurnal

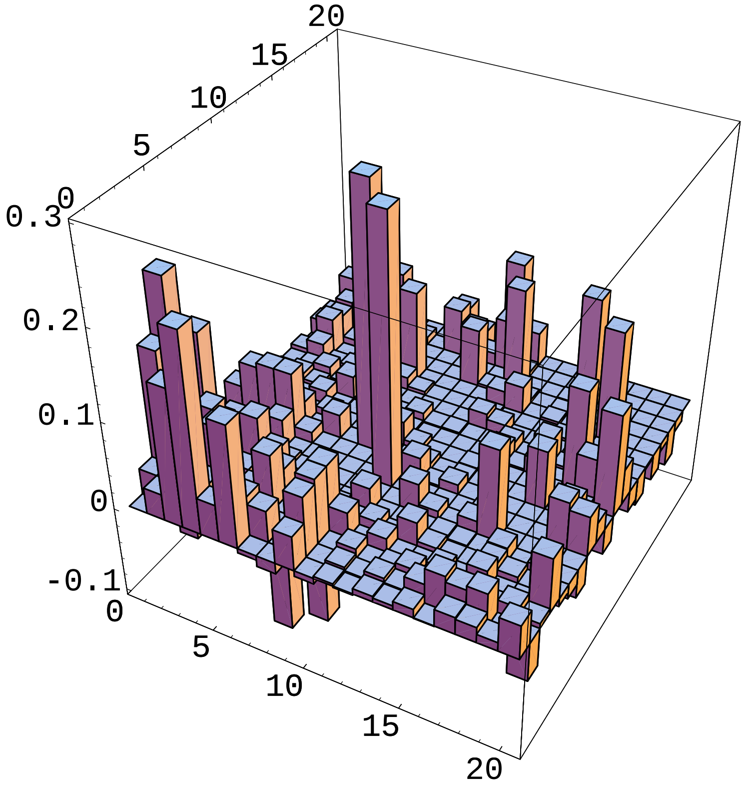


Figure 2.9: Correlation between 20 greatest OD pairs in Funet data

patterns, based only on the similar daily variation. Thus, it is more interesting to study the stochastic components to find out if there is any dependency between them.

To evaluate the dependency between OD pairs we have calculated cross-correlation between the residuals $z^{\Delta}_{n,k}$ and $z^{\Delta}_{n,k'}$ of different OD pairs $k$ and $k'$:

$$r(k,k') = \frac{\sum_{i=1}^{n}(z^{\Delta}_{i,k} - \bar{z}^{\Delta}_{k})(z^{\Delta}_{i,k'} - \bar{z}^{\Delta}_{k'})}{\sqrt{\sum_{i=1}^{n}(z^{\Delta}_{i,k} - \bar{z}^{\Delta}_{k})^2(z^{\Delta}_{i,k'} - \bar{z}^{\Delta}_{k'})^2}}.$$

For this purpose the 20 largest OD pairs are selected from the traffic matrix. The correlation values are depicted graphically in Figure 2.9, where the correlation terms of an OD pair with itself, which would obviously equal $1.0$, are left out. The distribution of the correlation terms is shown in Figure 2.10, where the vertical lines indicate the 95% confidence interval for the hypothesis that the correlation would be zero. That is, only $5\%$ of coefficients should fall outside that region, if the correlation would indeed be zero. Clearly there are a large number of statistically significant non-zero values in our data, although only a few have correlation larger than $0.1$.

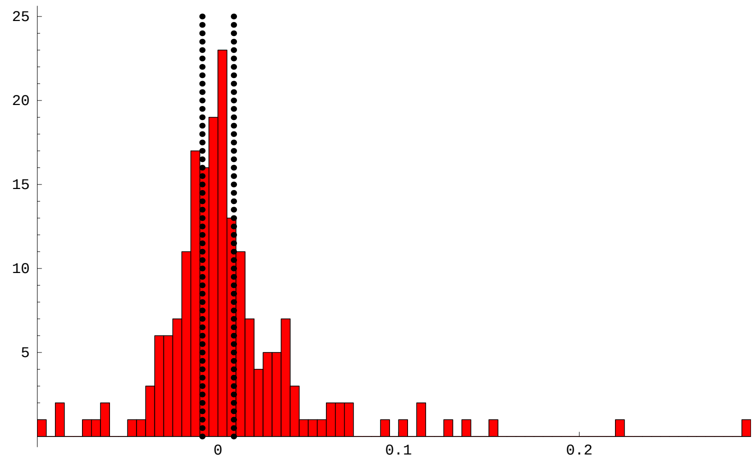


Figure 2.10: Distribution of correlation coefficients, with 95% confidence interval depicted by dotted lines.

We can conclude that strictly speaking the assumption of OD pair independence seems to be incorrect, but the observed dependencies are not very large.

To better understand the correlation between OD pairs, we concentrate on those OD pair couples that have the greatest correlation, either positive or negative. Table 2.1 lists 10 such pairs together with the origin and destination network for all OD pairs considered. This demonstrates that it is not only the pairs that share a common source (or destination) network that are correlated. That could be understood by the behavior of a source which has a bursty sending rate, so that it is sending traffic to many destinations at the same time, and then has an idle period, making the traffic from that node to destination nodes correlated. However, also OD pairs that have completely different origin and destination networks can have statistically significant correlation between them.

## 2.3 Static Routing

A typical simplifying assumption in traffic matrix estimation is to assume that the routing tables remain unchanged during the period of which we take measurements into consideration. Of course, in reality, routing in a network is unstable. Thus, in realistic situations we cannot necessary use too long collecting periods if we wish to use a single routing matrix for that entire time period.

Table 2.1: Origin and destination networks of pairs of OD pairs and cross-correlations between them

| $s_1$ | $d_1$ | $s_2$ | $d_2$ | $r$ |
|---|---|---|---|---|
| 65 | 42 | 51 | 36 | 0.29 |
| 1 | 5 | 1 | 2 | 0.22 |
| 1 | 4 | 1 | 2 | 0.15 |
| 1 | 21 | 1 | 2 | 0.14 |
| 30 | 5 | 66 | 5 | 0.13 |
| 30 | 46 | 30 | 4 | 0.11 |
| 30 | 5 | 34 | 1 | 0.11 |
| 30 | 5 | 23 | 35 | 0.10 |
| 66 | 5 | 23 | 35 | 0.10 |
| 51 | 36 | 1 | 3 | -0.10 |

### 2.3.1 Routing Stability in the Commercial Network

We study the routing table of the commercial network as a function of time to find out how long the stable periods are in reality. Figure 2.11 shows a trace of total traffic volume for a period of one week. The measurements are $10$ minute aggregates, and include the routing matrix for each $10$ minute period. The vertical lines in the figure represent a change in routing from the previous measurement period to the next. We note that there are long periods of static routing, and then in some occasions periods of high instability, where there are many consecutive changes in a short time period. In fact, for the most unstable moments, there were some $10$ minute periods that had several routing changes within the period, which are not depicted in the figure.

If we use for instance likelihood approach or the quick method of section 5, we can only consider collecting measurements from a period where the traffic is stationary. As we can clearly see from the figure, local stationarity in these cases is much more restricting than the static routing assumption. For a busy period of a few hours, it is likely that there will not be routing changes.

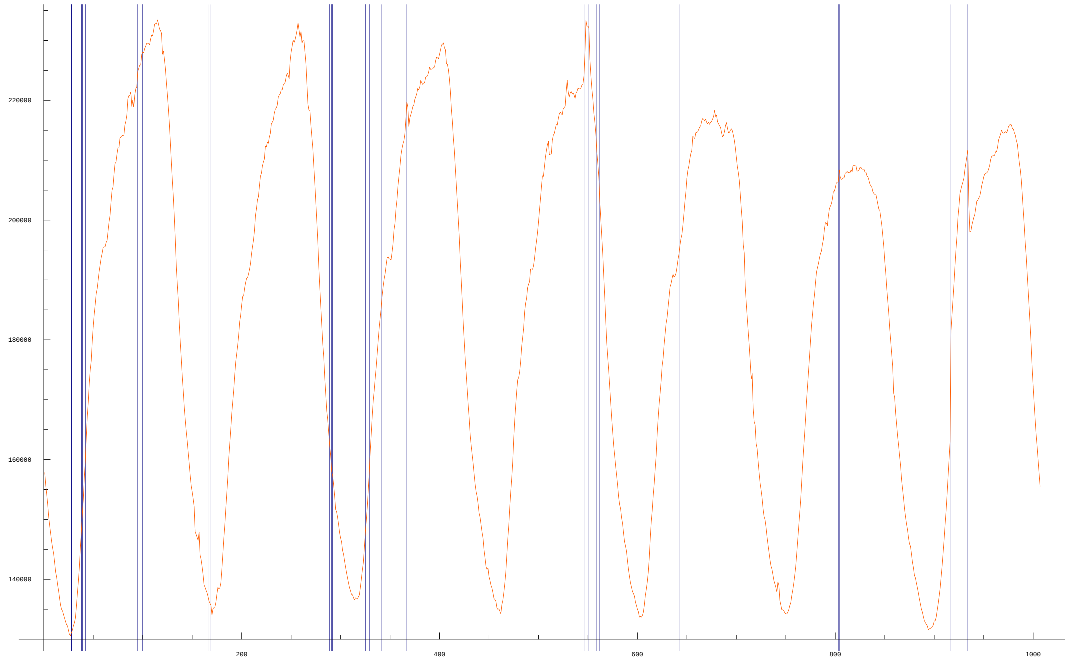


Figure 2.11: Routing changes in the commercial network

### 2.3.2 Random Routing in Poisson Models

However, if routing changes do occur, there are ways of handling this. Vardi [27] and Tebaldi [26] allow for a random Markovian routing, meaning that in each router the routing choice is independent of the path that the packet took to get there. Consider an example of Figure 2.12.

In this case we have four unidirectional links, and the OD pairs $x_{AB}$,$x_{BD}$,$x_{AC}$,$x_{CD}$ and $x_{AD}$. The first four can only use one link to reach the destination, but the last one has a choice between two routes: $A-B-D$ and $A-C-D$. Let us assume that each path is selected equally likely. Then the routing matrix would be

$$\boldsymbol{A} = \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 & 0.5 \\ 0 & 0 & 1 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0.5 \end{array}\right).$$

The idea is to divide the two routes into separate OD pairs, thus forming a so-called

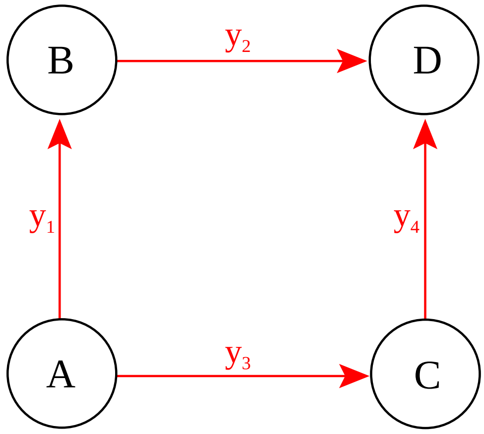


Figure 2.12: Example topology of random routing

super-network, for which the routing matrix is expanded into

$$\boldsymbol{A} = \left(\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right),$$

where the fifth column would be OD pair $x_{ABD}$ and the sixth column OD pair $x_{ACD}$. Because of the Markovian property, the likelihoods of each path can be easily calculated even for a more complicated network. Since we know the probability $p_i$ of each route, the expected value of the corresponding OD pair is just

$$\lambda_{ACD} = p_{ACD}\lambda_{AD}.$$

The Poisson model used in [26, 27] is thus structurally unchanged. After the traffic matrix of the super-network is estimated, the original traffic matrix is obtained by just summing the elements of OD pairs, in this case

$$\lambda_{AD} = \lambda_{ABD} + \lambda_{ACD}.$$

However, in order for the Poisson process to be valid for the new situation, the routing has to be truly random over the measurement collecting period. The data from the commercial network indicates that a more realistic situation would be that the routing changes are far apart and we have long stable periods in between.

### 2.3.3 Routing Changes in Maximum Likelihood Estimation

In section 3.4 we consider Maximum likelihood estimation (MLE). For the MLE the likelihood function is a product where each term corresponds to one measurement. Thus if we know when the routing changes occur, we can easily use different routing matrices for different measurement times without complicating the estimation.

## 2.4 Mean-Variance Relationship

To make the maximum likelihood approach considered in section 3.4 solvable, a common approach is to introduce a functional relationship between the mean and the variance of the traffic volumes. Vardi [27] and Tebaldi [26] in essence already did this by using the Poisson distribution, with the implicit assumption of variance being equal to the mean. Cao et al. [5] then generalized this to the Normal distribution by using a two-parameter power-law relation, where the variance is related to the mean through exponent parameter $c$ and scaling parameter $\phi$. This model has been widely adopted since.

$$\mathbf{\Sigma} = \phi \cdot \mathrm{diag}\{\boldsymbol{\lambda}^c\} = \phi \mathbf{\Lambda}^c, \tag{2.1}$$

where we denote $\mathbf{\Lambda}^c = \mathrm{diag}\{\boldsymbol{\lambda}^c\}$ for simplicity.

In this section we study this mean-variance relation. We make the separation between two different situations that are not to be confused with one another. First, by *temporal relation* we mean that the variance of a particular OD pair's traffic at a given time is related to the volume of the traffic at that time. That is, when there are more traffic, also the variation is higher. The second situation is the *spatial relation* in which we consider the relation over OD pairs. That is, it is studied whether the variance of an OD pair is larger for the OD pairs that have larger traffic volumes. The latter one is a key assumption in many traffic matrix estimation techniques, but we will give an overview of the temporal relation also, to clearly separate these two different situations.

The mean-variance relation is studied by dividing the data into non-overlapping periods and calculating the sample mean and sample variance for each of the periods. One hour periods are used here, but using different length periods yield similar results. For the temporal relation we then consider the set of these mean-variance pairs for a given OD pair over time. For the spatial relation the set considered is over all OD pairs for a given hour.

The logarithm of the mean-variance power law relation (2.1) is

$$\log \mathbf{\Sigma} = c \ \log \mathbf{\Lambda} + \log \phi.$$

Thus, in the log-log scale used in the figures, the exponent $c$ is a linear coefficient. If the relation holds, the points would fall on a line with slope $c$ and intercept $\log \phi$.

### 2.4.1 Temporal Relation

Regarding the temporal relation, Medina et al. [16] concluded that "for some OD pairs the Gaussian assumption may be just fine, but for others it does not work well." They observed that the exponent $c$ varies from one OD pair to another within bounds $c \in [0.5, 4.0]$. These observations were based on data collected from a Tier-1 backbone with measurement period $\Delta = 1$ s. Soule et al. [23] found similar results from flow data collected from a commercial Tier-1 backbone with measurement period $\Delta = 300$ s. They report that the power-law constant $c$ for individual OD flows varies in a range from $1$ to $4$. We found that this is true also for the Lucent local area network, examined in [5]. For some OD pairs the fit was reasonably good, but for others it was next to non-existent.

For the Funet data, we study the temporal mean-variance relation for the virtual OD pairs. We have selected four typical OD pairs that are among the largest in traffic volume but are different from each other regarding their statistical characteristics. The mean-variance pair for a given hour comprises one point in the plots of Figure 2.13 and 2.14, which thus have $24$ points.

Figure 2.13 depicts the situation when the measurement interval is one second, and Figure 2.14 when it is $60$ seconds. The best linear fit in the least square sense is depicted in each of the figures. The only OD pair out of the four that shows behavior that would indicate the existence of such relation is the one on the top right corner of the figures, which has a goodness of fit value of $R^2 = 0.92$ for the mean-variance relation, and estimated parameter value of $c = 1.9$. For the other OD pairs the fits are much worse. Changing the measurement intervals made the fit worse even for this OD pair, with $R^2 = 0.76$ and $c = 2.9$. For the other pairs the changed measurement interval changes the $c$-parameter even more dramatically. This is due to the fact that even if there is no relation, some parameter value always gives the best fit, and it may change significantly even due to a small change in the data.

In general, there does not seem to be a temporal mean-variance relationship in our data. However, this is not crucial for traffic matrix estimation. The spatial relation considered in the following section is the one we are more interested in.

### 2.4.2 Spatial Relation

In Cao et al. [5] the authors consider only integer values for the spatial mean-variance relation and conclude that $c = 2$ gives reasonably good fit. Our study [1] of the same local area network data set yielded parameter value of $c = 1.96$, as shown in Figure 2.15. Medina et al. [16] reported that the mean-variance relationship seems to hold over all OD pairs with the parameter $c = 1.97$. Gunnar et al. [12] also confirmed the validity of the relation (during the busy hour over all OD pairs) and reported parameter values $c = 1.5$ and $1.6$ based on data traces from a global operator's backbones in US and Europe. Soule et al. [23] found that over all OD pairs in a backbone network the best linear fit resulted in value $c = 1.56$.

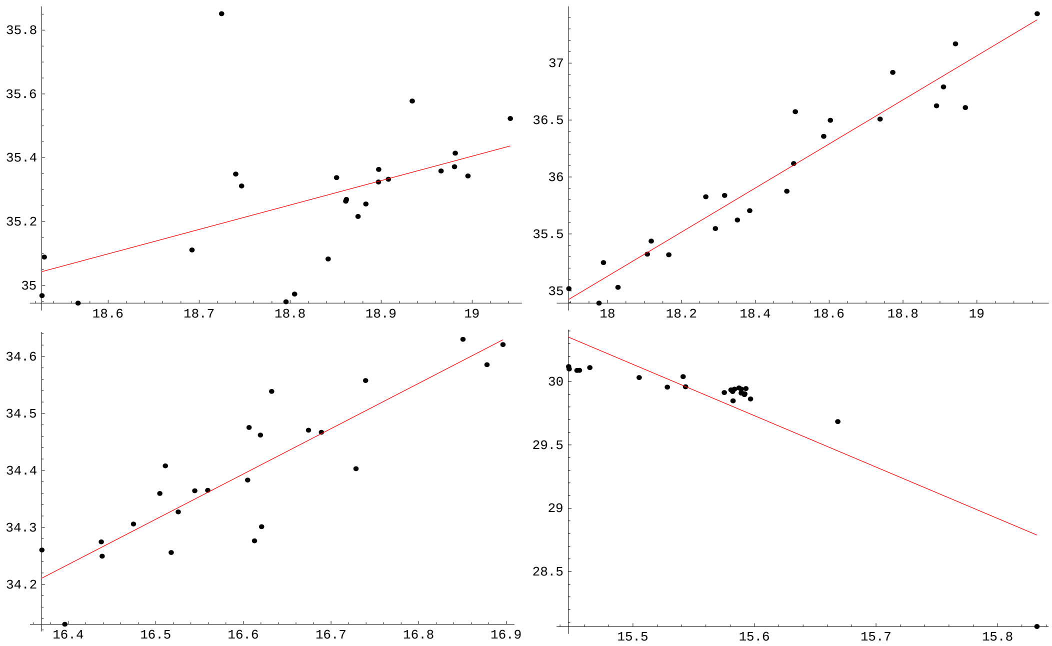


Figure 2.13: Temporal mean-variance relation for Funet virtual OD pairs on one second resolution.

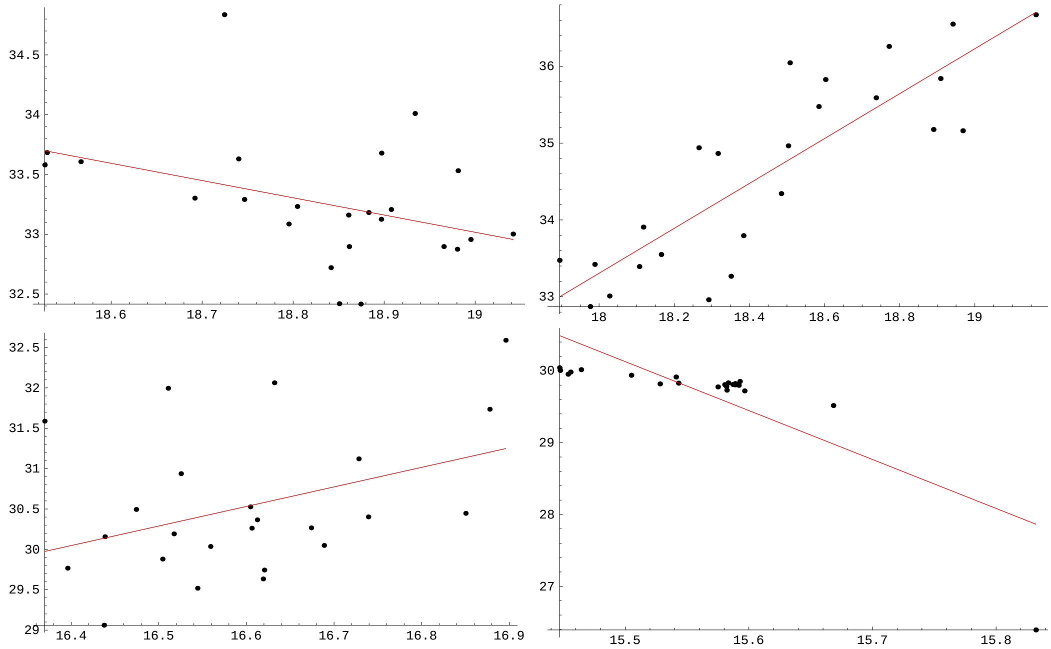

Figure 2.14: Temporal mean-variance relation for Funet virtual OD pairs on $60$ second resolution.

However, they did not find the fit sufficiently good to justify the use of the relation, although they remain uncertain as to the effect this inaccuracy brings to the estimation results. We study this below through a simulation study. Also, they express concerns that the varying parameter values of temporal relation might hinder the use of the spatial relation in traffic matrix estimation. While it is true that reasons that cause the temporal relation not to hold might also cause the spatial relation to be inaccurate, clearly this is not always the case. As we will see below, the spatial relation holds considerably better for the Funet data set than the temporal relation, which did not hold at all.

We study the spatial relation between the $40$ largest virtual OD pairs of the Funet data set, concentrating on one-hour periods that we assume approximately locally stationary. The average goodness of fit is $R^2 = 0.83$ with the values ranging from $0.60$ to $0.95$, while the estimate for the exponent parameter vary from $1.11$ to $1.46$, with $1.34$ being the average. For some hours the fit is reasonably good, as shown in Figure 2.16, where one point in the plot depicts the mean and variance of one OD pair during the given hour. For some other one-hour periods the fit is not very good. Changing the measurement interval to $60$ seconds or taking more OD pairs into consideration does not affect the situation significantly.

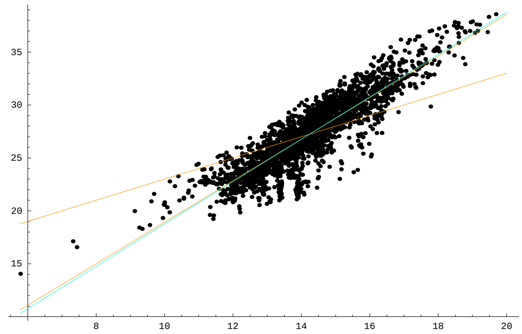


Figure 2.15: Spatial mean-variance relation of the Lucent data set.

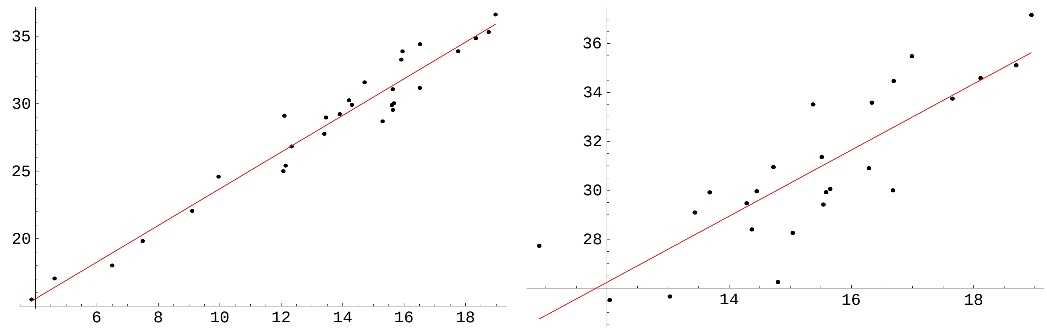


Figure 2.16: Spatial mean-variance relation of the Funet data set, examples of good and not so good fits. Left: $R^2 = 0.95$, right: $R^2 = 0.70$.

We can conclude that for our OD pairs, there is a vague spatial mean-variance relation, with good fits for some one hour periods. It is to be noted however, that we have an extremely high resolution in dividing the trace into these virtual OD pairs, so the situation is different from the typical traffic matrix estimation situation, where the OD pairs are larger and thus more aggregated, as in the study by Gunnar et al. for instance. This might affect also the validity of the mean-variance relation.

### Effects of Inaccuracies in the mean-variance Relation

An important aspect regarding what we can conclude about the validity of the mean-variance relation is to study the actual effect inaccuracies cause to estimates. We study this through a simulation, using a six node topology, with $30$ OD pairs. Synthetic data sets were created, where the mean-variance relationship holds to different degrees. For each goodness of fit value we performed the simulation several times by drawing new set of synthetic Gaussian measurements of sample size $100$, with

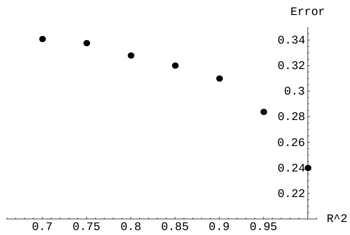


Figure 2.17: Errors in traffic matrix estimation as a function of the goodness of fit of the mean-variance relation.

the underlying parameters staying the same. After obtaining maximum likelihood estimates by the EM algorithm, discussed in section 3.4.4, the average error of the estimates is then computed for each scenario.

The results of the errors as a function of the goodness of fit value for the mean-variance relation used in the simulations can be seen in Figure 2.17. The effect of a bad fit is not as dramatic as one might think. Even with $R^2 = 0.70$ the errors are less than $1.5$ times as large as in the ideal situation, This is due to the fact that maximum likelihood estimates are dominated by the first order equation, and the mean-variance relation is used only to get the second order terms to bring in the extra information needed to make the problem identifiable.

The reasonable accuracy is all the more surprising, when remembering that the $R^2$-values are computed on the log-log scale. Although a $0.70$ goodness of fit, as depicted in Figure 2.16, looks reasonably good on that scale, on the linear scale the accuracy is not very good.

However, we must note that at values close to $1.0$ the situation is quickly deteriorating as the fit grows worse. In our data set the average fit was $R^2 = 0.83$. Around that kind of values a change of $0.05$ in the goodness of fit to either direction is not too critical, but there is a clear price to be paid in estimation accuracy for the fact that the relation does not hold exactly.

# Chapter 3

# Review of Estimation Methods

*In this chapter we take a comprehensive look on various estimation techniques proposed in the literature. These can be roughly divided into a few main groups based on the nature of the extra information brought in to make the problem solvable. In the Bayesian methods there is an assumption about a prior distribution, which may be obtained by previous knowledge of the traffic matrix or some other type of estimation technique. In the Maximum likelihood methods the link covariances are used to obtain a likelihood surface with an unambiguous maximum point. This calls for the use of the mean-variance relationship. The gravity based methods assume that the traffic between two nodes is proportional to the product of the total traffic volumes of the nodes.*

## 3.1 Introduction

Even though it is agreed that the knowledge of accurate traffic matrices is crucial, for example, for traffic engineering, for a long time there were only a few proposals for methods to obtain the traffic matrices from information available in a typical IP network. While in recent years more methods have been proposed, it is interesting to notice how strikingly few different approaches there really are. We have listed the current methods in Table 3.1, along with the number of the section where the method in question is explained in detail.

Table 3.1: Taxonomy of estimation methods

| **Method** | **Section** | **Source of extra information** |
|---|---|---|
| Simple Gravity model<br>Kowalski and Warfield[13] | 3.2.1 | Gravity model |
| Generalized Gravity model<br>Zhang et al. [30] | 3.2.2 | Gravity model |
| Choice model<br>Medina et al. [16] | 3.2.3 | Mlogit Gravity model |
| Constant Fanout model<br>Gunnar et al. [12] | 3.2.4 | Constant fanouts |
| Tomogravity<br>Zhang et al. [30] | 3.2.5 | Gravity model |
| Information Theoretic approach<br>Donoho et al. [31] | 3.2.6 | Gravity model |
| Bayesian Inference<br>Tebaldi and West[26] | 3.3.1 | Known prior distribution<br>Poisson distribution |
| Iterative Bayesian Inference<br>Vaton and Gravey [28] | 3.3.2 | Known prior distribution<br>Poisson distibution |
| Network tomography<br>Vardi [27] | 3.4.2 | Mean-variance relation (Poisson)<br>Poisson distribution |
| Time Varying Network tomography<br>Cao et al. [5] | 3.4.5 | Mean-variance relation (general)<br>Normal distribution |
| Scalable likelihood approach<br>Cao et al. [6] | 3.4.6 | Mean-variance relation ($c = 1$)<br>Normal distribution |
| Pseudo Likelihood estimation<br>Liang and Yu [15] | 3.4.7 | Mean-variance relation (general)<br>Normal distribution |
| Quick Method<br>Juva et al. [3] | 5 | Mean-variance relation (general)<br>Normal distribution |
| Linear Programming<br>Goldschmidt[11], Eum et al. [10] | 3.5.1 | None (Interior point method<br>initialization as "prior") |
| Worst case bounds by LP<br>Gunnar et al. [12] | 3.5.1 | None |
| Route Change Method<br>Soule et al. [23] | 3.5.2 | Ability to change link weights<br>at specified moments |

The key element in making an estimate in an ill-posed situation is the source of the extra information brought in to make the problem identifiable. For each method listed in the table, the nature of this information is given in the right column. The two most common approaches by far are the gravity model and its extensions like the constant fanout hypothesis, and the mean-variance relation. The role of the extra information is in essence to single out a solution out of the subspace defined by the link count equations. The proposed techniques are just ways to combine the link count measurements and the extra information to produce an estimate for the traffic matrix.

The gravity model based methods are reviewed in section 3.2. The Bayesian methods do not specifically have to use the gravity model, but they do need a prior distribution for the traffic matrix and gravity model is a natural candidate, unless an outdated yet somewhat accurate version of the traffic matrix is available. We will present them in section 3.3.

The mean-variance relation was studied in section 2.4, where we mentioned that it is a crucial assumption for a number of estimation techniques. It is used in the maximum likelihood framework, to make the likelihood surface have a unambiguous maximum point. In the quick method (section 5) we propose to use it as a basis of a closed form solution.

In the light of the fact that some extra information is always needed in an ill-posed situation, the attention is drawn to the LP method in the table, which does not use any extra information, other than preselection of weights in the objective function. The worst case bound method does not yield an estimate, only bounds for the possible values of each element in the traffic matrix, so it does not require extra information. The LP methods of [11] and [10] on the other hand are either revolutionary breakthroughs or just don't work. We argue for the latter in section 3.5.1.

## 3.2 Gravity Model and Extensions

The gravity model is named after Newton's law of gravitation. As in the law of gravitation the force between two objects is proportional to the masses of the objects and the inverse of the square of the distance between them

$$F \propto \frac{m_1 m_2}{r^2},$$

similarly in the gravity modelling for data networks the traffic between two nodes is assumed to be proportional to the total traffic volumes of those nodes. Gravity models have been used in social science to model the movement of people or goods between two areas, as well as in telephone networks. The idea is that if we have no knowledge of where a bit is coming or where it is going, the best guess is to make the estimate proportional to traffic volumes sent and received by each node in the network.

### 3.2.1 Basic Gravity Method

The general form of the gravitation model has a repulsion term and an attraction term that are multiplied together and then divided by a distance function. In the case of traffic matrix estimation it can be written in the form proposed by Kowalski and Warfield [13] for teletraffic demands:

$$X_{sd} = k_s \frac{O_s T_d}{d_{sd}^{\alpha_s}}. \tag{3.1}$$

The repulsion term is $O_s$ which is the total traffic originating from node $s$. The attraction term is $T_d$, the total traffic terminating at node $d$. The numerator $d_{sd}$ is a distance function between nodes $s$ and $d$, where $\alpha_s$ is the distance parameter. Coefficient $k_s$ is a normalizing constant.

Zhang et al. [30] use an approach where the normalizing coefficient and distance function are put together to form the friction factor $f_{sd}$ between origin and destination,

$$X_{sd} = \frac{O_s T_d}{f_{sd}}. \tag{3.2}$$

However, they notice that the inference of the friction factors is an equivalent problem to the traffic matric inference, and thus ill-posed. Hence, the factors have to be estimated using fewer parameters. The authors drop the distance function altogether and use a constant for $f_{sd}$ reducing it to only normalizing coefficient for the results given by the denominator in (3.2). The equation is then

$$X_{sd} = k \cdot T_s^{in} T_d^{out}, \tag{3.3}$$

where $T_s^{in/out}$ is the amount of traffic entering/leaving the network through node $s$ and the normalizing constant is

$$k = \sum_i T_i^{out} \qquad \text{or} \qquad k = \sum_i T_i^{in},$$

with both yielding identical results.

The authors find that this simplification yields surprisingly accurate results. Namely, accurate enough for the intended initialization use.

### 3.2.2 Generalized Gravity Model

Zhang et al. [30] generalize the gravity model to handle additional information, specifically to use the knowledge that some of the egress links are peering links to other ISPs while others are access links, to differentiate between customer and peering traffic. Access links are denoted by $a$, peers by $P$ and peering links by $p$. The generalized gravity model considers link to link traffic matrices. The situation is depicted in Figure 3.1, where the topology has two peers with several peering links and many customers connected to the backbone by access links.

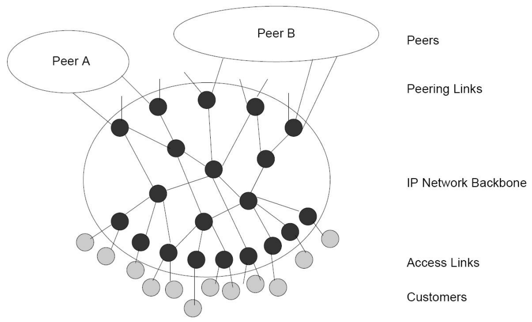


Figure 3.1: Example topology for generalized gravity model [30]

The authors assume that transit traffic between peering links is negligible. Each peer $P_j$ has several peering links $p_m$, but all traffic from an access link to specific peer will use the same peering link. The peering link for access link $a_i$ is denoted by $X(a_i, P_j)$.

We have three separate cases, each following the gravity model.

- Outbound traffic from access link to peering link.
  - If the considered link $p_m = X(a_i, P_j)$, the traffic is proportional to traffic terminating at peer $P_j$ and to traffic originating from link $a_i$.
- Inbound traffic from peering link to access link
  - Proportional to traffic entering the network through peering link $p_i$ and to traffic exiting the network through access link $a_j$.
- Internal traffic between access links.

– Proportional to traffic entering the network through peering link $a_i$ and to internal traffic exiting the network through access link $a_j$, meaning that the traffic that comes from peering links is not included.

The generalized equations are as follows

$$T_{outbound}(a_i, p_m) = \frac{T_{link}^{in}(a_i)}{\sum_{a_k \in \mathcal{A}} T_{link}^{in}(a_k)} T_{peer}^{out}(P_j), \quad \text{if } p_m = X(a_i, P_j), \tag{3.4}$$

$$T_{inbound}(p_i, a_j) = \frac{T_{link}^{out}(a_j)}{\sum_{a_k \in \mathcal{A}} T_{link}^{out}(a_k)} T_{link}^{in}(p_i), \tag{3.5}$$

$$T_{internal}(a_i, a_j) = \frac{T_{link}^{in}(a_i)}{\sum_{a_k \in \mathcal{A}} T_{link}^{in}(a_k)} T_{internal}^{out}(a_j), \tag{3.6}$$

where $T_{internal}^{out}(a_j)$ is the portion of total traffic $T^{out}(a_j)$ out of $a_j$, that is not going to any of the peering links.

### 3.2.3 Choice Model

Medina at al. [16, 18, 19] introduce the choice model for POP to POP traffic matrix estimation, where they combine the attraction term of the destination node with the distance function to form a fanout term $\alpha_{sd}$ that determines which portion of the traffic from a source node $s$ is going to each destination node $d$. The choice model is thus written as

$$X_{sd} = O_s \alpha_{sd}. \tag{3.7}$$

Then, to estimate the fanouts, a Discrete Choice Model (DCM) is used. The estimation problem reduces to estimating the parameters of the DCM model. This can be understood so that each ingress POP is a decision maker that makes the choice of which egress POP it sends a packet. While the POPs are obviously not intelligent agents that would make choices themselves, it can be interpreted that way by combining to each POP the user behavior and the choices made in the network design.

The decision process is modelled by the utility maximization criterion. The utility function is given as a weighted sum of different attributes $m$ affecting the attractiveness of choice of node $d$ to node $s$.

$$V_d^s = \sum_m \mu_m w_d^s(m) + \gamma_d,$$

where the $w$:s are the attributes, $\mu_m$ is the weight given to attribute $m$, and $\gamma$ is a scaling term. The authors propose two models. The first one is a model with a single attribute $w_d$ that is the total amount of traffic coming into the egress node. The other one has two attributes, adding attribute $w^s$, that gives the total amount of traffic leaving the ingress node. Both of these are obtained by summing the link counts on the links adjacent to source and destination nodes, respectively.

The authors conclude that the two attribute model

$$\begin{aligned} V_d^s &= \mu_1 w_d^s(1) + \mu_2 w_d^s(2) + \gamma_d &(3.8)\\ &= \mu_1 w_d(1) + \mu_2 w^s(2) + \gamma_d &(3.9)\end{aligned}$$

works better in their test scenarios.

The probability of node $s$ choosing node $d$ is then modelled as a multinomial logit model

$$\alpha_{sd} = P_d^s = \frac{e^{V_d^s}}{\sum_k e^{V_k^s}}.$$

These probabilities are used to model the fanout terms $\alpha_{sd}$. Thus the traffic between $s$ and $d$ is given as

$$X_{sd} = O_s \frac{e^{V_d^s}}{\sum_k e^{V_k^s}}. \tag{3.10}$$

In Figure 3.2 the choice model is compared to gravity model in a study with synthetic traffic. It is found that the choice model yields more accurate estimates. This is expected as the model has parameters that need to be calibrated, as opposed to gravity model which does not have free parameters.

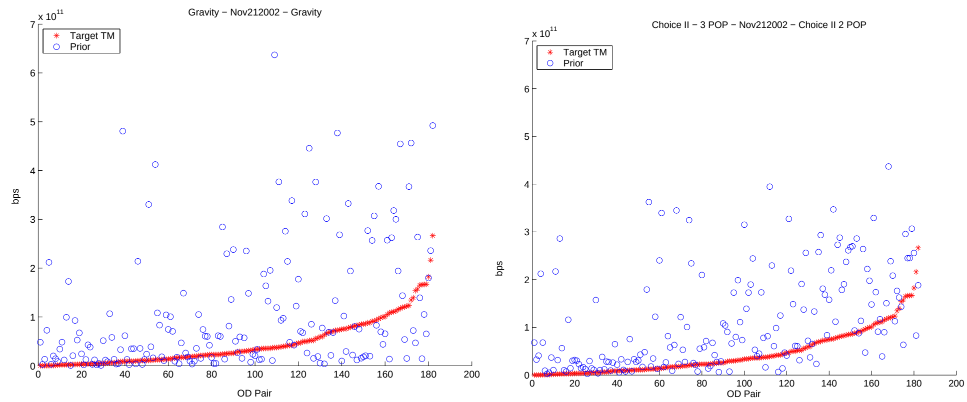


Figure 3.2: Prior distribution comparison with synthetic traffic. Left: Gravity prior, right: Choice model prior. [18]

So the problem is still to find values for the parameters $\mu_1, \mu_2$ and $\gamma_d$. This cannot be done accurately without partial direct measurement of the traffic matrix being available. This would require having packet traces or Netflow measurements from a few links. The linear choice model

$$\alpha_{sd} = P_d^s = \frac{V_d^s}{\sum_k V_k^s},$$

where the weights are set to $\mu = 1$ does not require calibration. The single attribute model is now equivalent to the simple gravity model, and the two-attribute model is an extension of the gravity model.

### 3.2.4 Constant Fanout Model

Gunnar et al. [12] analyze real traffic from the Global Crossing backbone network. They find that for the European network the simple gravity model is reasonably accurate, but for the American network it is not. In that case the gravity model significantly underestimates the traffic of the largest OD pairs. However, they find that the fanout factors $\alpha$ remain constant over time, even while the traffic amounts fluctuates due to a diurnal pattern. The fanout $\alpha_{sd}$ gives the percentage of traffic that source node $s$ sends to destination node $d$ of its total traffic. Thus the fanouts sum to unity for each source node.

$$\sum_d \alpha_{sd} = 1, \qquad \forall s.$$

Assuming the constant fanouts they write

$$\boldsymbol{x}_t = S_t \boldsymbol{\alpha},$$

where $S_t$ is a time dependent scaling term and $\boldsymbol{\alpha}$ is the vector of fanout terms, which sum to unity for each source node. Thus the link count equations get the form

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{A} S_t \boldsymbol{\alpha} \\ \sum_d \alpha_{sd} &= 1 \qquad \forall s. \end{aligned} \tag{3.11}$$

With a time series of link counts, the above system will quickly become overdetermined, and the optimization problem

$$\begin{aligned} \min \qquad & \sum_t ||\boldsymbol{A} S_t \boldsymbol{\alpha} - \boldsymbol{y}_t||_2^2 \\ \text{subject to} \qquad & \sum_d \alpha_{sd} = 1 \qquad \forall s \end{aligned}$$

will have an unique solution vector $\boldsymbol{\alpha}$.

### 3.2.5 Tomogravity

A major drawback of the gravity model is that it does not utilize the available link count information. The solution usually does not satisfy the link count equation

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x},$$

and can be thus improved upon by incorporating this into the estimate. Zhang et al. [30] use the (generalized) gravity model to obtain a starting point $\boldsymbol{x}_0$, and solve the quadratic programming problem of the $L_2$ norm of a vector, e.g., the euclidian distance

$$\begin{aligned} \min \quad & ||(\boldsymbol{x} - \boldsymbol{x}_o)/\boldsymbol{w}|| \\ \text{so that} \quad & ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}|| = 0, \end{aligned} \tag{3.12}$$

where $\boldsymbol{w}$ is a weight vector, and the division is performed componentwise. Constant weight vector leads to the least square solution yielding the point on the link count condition sub-space that is closest to the prior gravity solution. Weighted least squares lead to different solution. By giving a large weight to an OD pair means that the estimate for this OD pair will change more in the procedure of projecting the gravity estimate to the link count condition sub-space, while smaller weight means that it "costs" more with regard to the objective function to move away from the prior in the direction of that particular OD pair. The weight setting has similar effect as the covariance of the prior in Bayesian approach. We study this further in section 3.3.3.

Finally, it is possible that due to an inaccurate prior, the least squares method will yield negative values. These are handled by an iterative proportional fitting algorithm to ensure a non-negative solution.

### 3.2.6 Information Theoretic Approach

In [31] the tomogravity method is generalized using an information theoretic approach. The gravity model is based on independence between origin and destination of the traffic. In information theoretic terms this can be expressed by the mutual information $I(S, D)$ between source and destination addresses, where $S$ and $D$ are random variables with have values $s$ and $d$ for a specific source $s$ and destination $d$.

The mutual information can be expressed in many different ways, but the most useful interpretation for this problem is

$$I(S, D) = K(p(s, d)||p(s)p(d)),$$

where

$$K(f||g) = \sum_i f_i \log\left(\frac{f_i}{g_i}\right),$$

which is the Kullback-Leibler divergence measuring the distance between distributions $f$ and $g$. So we can write

$$I(S, D) = \sum_{s,d} p(s, d) \log_2\left(\frac{p(s, d)}{p(s)p(d)}\right).$$

The authors note that a typical way of solving ill-posed linear inverse problems is to solve the regularized minimization problem with a penalty function. In this case

$$\min_{\boldsymbol{x}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2^2 + \lambda^2 J(\boldsymbol{x}), \tag{3.13}$$

where $\lambda$ is a regularization parameter and $J$ is a penalization functional. For instance the Bayesian approach of Tebaldi et al. (see section 3.3.1) can be written in this form, by choosing $J(\boldsymbol{x}) = \log \pi(\boldsymbol{x})$, where $\pi(\boldsymbol{x})$ is the prior distribution for $\boldsymbol{x}$.

The authors use probabilistic terms in their notation of the problem. Total traffic in the network is denoted by $N$, and the traffic sent from source $s$ to destination $d$ is denoted by $N(s, d)$. Thus

$$N(s, d) = Np(s, d),$$

where $p(s, d)$ is the probability that a random bit in the network goes from node $s$ to node $d$. The OD pairs are indexed by $i$, and the origin and destination of the $i$th OD pair are denoted by $s_i$ and $d_i$. The gravity estimate $g_i$ for the OD pair's traffic is defined as the product of all traffic originating from $s_i$ and all traffic terminating at $d_i$.

$$g_i = p(s_i)p(d_i) = N(s_i)N(d_i)/N^2,$$

and

$$f_i = p(s_i, d_i) = N(s_i, d_i)/N = x_i/N.$$

In information theoretic terms the independence between source and destination, implied by the gravity model, is equivalent to the mutual information being zero.

As the mutual information $I(s_i, d_i)$ is also always positive, it is thus an appropriate penalty function.

$$J(\boldsymbol{x}) = I(s_i, d_i) = \sum_i f_i \log\left(\frac{f_i}{g_i}\right) = \sum_i \frac{x_i}{N} \log\left(\frac{x_i/N}{g_i}\right).$$

Now equation 3.13 can be written as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||^2 + \lambda^2 \sum_{i:\ g_i > 0} \frac{x_i}{N} \log\left(\frac{x_i/N}{g_i}\right) \\ \text{subject to} \quad & x_i \geq 0. \end{aligned} \tag{3.14}$$

That is, we want a solution that is a tradeoff between satisfying the link count relation and having an a priori plausibility, which here means that the mutual information is small and the solution is thus close to the gravity model. The final solution depends on the selection of $\lambda$. The authors use value $\lambda = 0.01$, but demonstrate that the accuracy of the method is not very sensitive to the choice of $\lambda$.

An equivalent formulation for the problem is

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & K(f||g) \\ \text{subject to} \quad & ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||^2 \leq \chi^2, \end{aligned}$$

where $\chi^2$ is a function of the selected $\lambda$. Using a linear approximation for the logarithm yields

$$K(f||g) \approx \sum_x \left[\frac{f - g}{\sqrt{g}}\right]^2. \tag{3.15}$$

The objective function (3.15) is of the type of the $L_2$ norm of (3.12). This shows that the tomogravity method with a square root weight function is an approximation of the information theoretic method. And indeed, root square weights were found to be most effective in tomogravity method.

## 3.3 Bayesian Approach

The Bayesian methods can be classified as a group of their own, although the fundamental difference to tomogravity, for example, is really only in the computational techniques. In principle there is not much of a difference. Bayesian approaches typically do not specify how to obtain the necessary prior distribution, but gravity model would be the obvious choice, leading to a situation where the Bayesian inference techniques incorportate the link count information to a gravity prior. which is exactly the idea behind tomogravity and information theoretic methods as well.

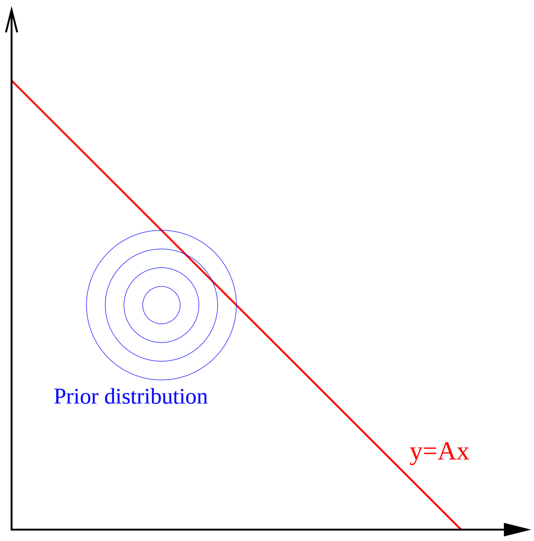


Figure 3.3: Starting point of Bayesian inference approach

### 3.3.1 Bayesian Inference

In the Bayesian approach the idea is to compute conditional probability distribution for OD pair traffic $\boldsymbol{x}$, given the link counts $\boldsymbol{y}$ and a prior distribution. The basic situation is shown in Figure 3.3 where the prior distribution is depicted by the circles corresponding to multiples of standard deviation, and the solution subspace is the line defined by the link count information.

In [26] a Poisson-distribution is used, and

$$X_n \sim \text{Poisson}(\lambda_i) \tag{3.16}$$

independently for all OD pairs $i$. The goal is to obtain

$$P(\boldsymbol{x}, \boldsymbol{\lambda}|\boldsymbol{y}),$$

the joint distribution of $\boldsymbol{X}$ and $\boldsymbol{\lambda}$ conditioned on the observed link counts $\boldsymbol{y}$.

Analytical computations are difficult in this case, and thus Markov Chain Monte Carlo (see e.g. [35]) methods are used to obtain the posterior distribution. The basic iteration is a standard Gibbs sampling, where a step is defined as

$$\begin{aligned} \boldsymbol{\lambda}^i &= P(\boldsymbol{\lambda}|\boldsymbol{x}^i, \boldsymbol{y}), & (3.17)\\ \boldsymbol{x}^{i+1} &= P(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{y}). & (3.18) \end{aligned}$$

And then iterate until feasible solution is found.

Since the traffic matrix $\boldsymbol{x}$ and routing matrix $\boldsymbol{A}$ can be partitioned as

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}, \qquad \boldsymbol{A} = (\boldsymbol{A}_1, \boldsymbol{A}_2),$$

so that $\boldsymbol{A}_1$ is invertible, it follows from (1.2) that

$$\boldsymbol{x}_1 = \boldsymbol{A}_1^{-1}(\boldsymbol{y} - \boldsymbol{A}_2\,\boldsymbol{x}_2). \tag{3.19}$$

So it suffices to compute only $P(\boldsymbol{x}_2|\boldsymbol{\lambda}^i, \boldsymbol{Y})$ in the iteration, after which $x_1$ is obtained from (3.19).

It is clear that a prior for $\boldsymbol{x}$ is needed to get the iteration started. Also, since the final solution combines the prior and the link count information, it is clear that the prior has a big effect on the final solution. For instance, the authors note that uniform priors may lead to gross over-estimation of low rates, and vice versa, under-estimation of higher rates. Indeed, good priors are absolutely crucial for the accuracy of this method, and this is also a major weakness of the Bayesian approach.

### 3.3.2 Iterative Bayesian Method

Vaton and Gravey make use of several successive link count measurements in their iterative Bayesian method [28] that allows for modulated process for the underlying traffic matrix distribution. The method consists of iteration and exchange of information between two "boxes" as depicted in Figure 3.4. The first box follows the method by Tebaldi, and simulates the traffic matrix from the link counts at each fixed time period using MCMC methods, and utilizing some prior distribution. On the first iteration the prior is obtained by the gravity model.

The first box's output consists of the estimated traffic matrices separately for each time period. For example, consider a toy example of three OD pairs and five mea-

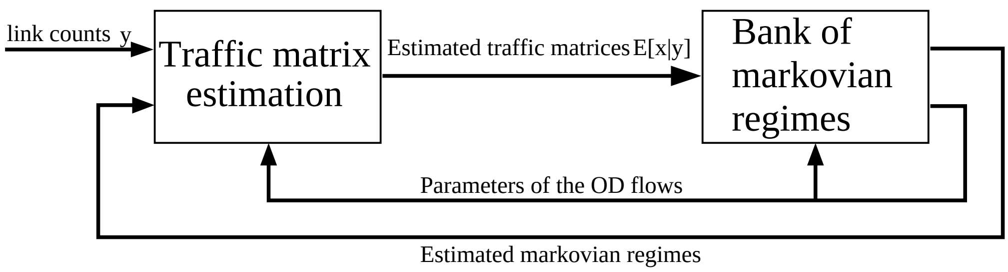


Figure 3.4: The Vaton-Gravey iterative method

Table 3.2: Illustration of Iterative Bayesian table of traffic matrices

| | | | | | |
|---|---|---|---|---|---|
| OD pair 1 | $x_{1,1}$ | $x_{1,2}$ | $x_{1,3}$ | $x_{1,4}$ | $x_{1,5}$ |
| OD pair 2 | $x_{2,1}$ | $x_{2,2}$ | $x_{2,3}$ | $x_{2,4}$ | $x_{2,5}$ |
| OD pair 3 | $x_{3,1}$ | $x_{3,2}$ | $x_{3,3}$ | $x_{3,4}$ | $x_{3,5}$ |

surement periods. The values are depicted in table 3.2, where in $x_{i,t}$ the subscript $i$ is the number of OD pair in question, and $t$ refers to the time slot in question. One column of the table, denoted by $\boldsymbol{x}_t$ is thus the traffic matrix for time slot $t$. One row of the table is a time series of values for the $i$th OD pair. In the first box the table is filled column by column, one time slot at a time.

The estimated traffic matrices are then given to the second box, which instead goes through the table row by row. The values of each row are fitted to the underlying model and the parameters for that model are estimated using maximum likelihood estimation. The updated estimates of the traffic matrices are then inserted back to the first box as a new prior distribution.

The authors demonstrate a case where the traffic counts for each OD pair constitute a Markov modulated process. In that case the second box updates the parameters of the process using the EM algorithm. These parameters are then fed back into the first box as a Bayesian prior, and the process is repeated, until the iteration converges.

### 3.3.3 Bayesian Methods under Gaussian Distribution

The methods presented in the two preceding sections both use the Poisson distribution for the traffic matrix. However, there are several reasons to consider using the Gaussian distribution instead.

1. Internet traffic has more variation than the Poisson assumption allows. The Gaussian model gives more leeway in modelling that extra variation.

2. Even if the Poisson assumption would hold, Gaussian distribution is a good approximation for the Poisson distribution.

3. For the Gaussian model we do not need to use the tedious MCMC simulation.

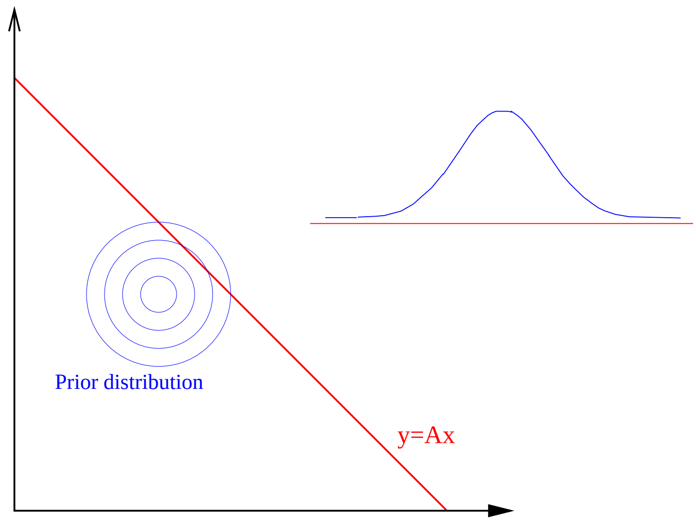


Figure 3.5: Gaussian prior yields a Gaussian posterior

The first point we showed in section 2.4, where traffic of the Finnish University network was analyzed and results from literature were discussed. The second point follows from the central limit theorem (see e.g. [34]), since if the Poisson model is true for packet arrivals, the five minute aggregates have a sufficiently large amounts of arrivals that the traffic volume is close to Gaussian. The final point follows from the fact that it is possible to obtain a closed form solution for the posterior in the Gaussian case, as we showed in [4]. As the prior is Gaussian, also the posterior is a Gaussian along the regression line given by the link count condition, as illustrated in Figure 3.5.

As the MCMC simulations are very time consuming, the closed form equation saves considerable computation time. Our solution started from the notation of Tebaldi, making use of equation (3.19). This leads to quite complicated equation of several matrices. However, there is a much simpler formulation for the conditional posterior. This is given by Cao et al. [5] as a byproduct of the Expectation Maximization algorithm.

$$\boldsymbol{\lambda} = \mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t] = \boldsymbol{\lambda}_0 + \boldsymbol{\Sigma}_0\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}_0\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\lambda}_0), \tag{3.20}$$

where the mean and covariance matrix of the prior distribution are denoted by $(\boldsymbol{\lambda}_0, \boldsymbol{\Sigma}_0)$ . The derivation of this expression, with the notation used in (3.32), is given in appendix A.

If we do not have prior knowledge of the covariance matrix, we can assume it to be a diagonal with the same value for each term. This simplifies the above equation

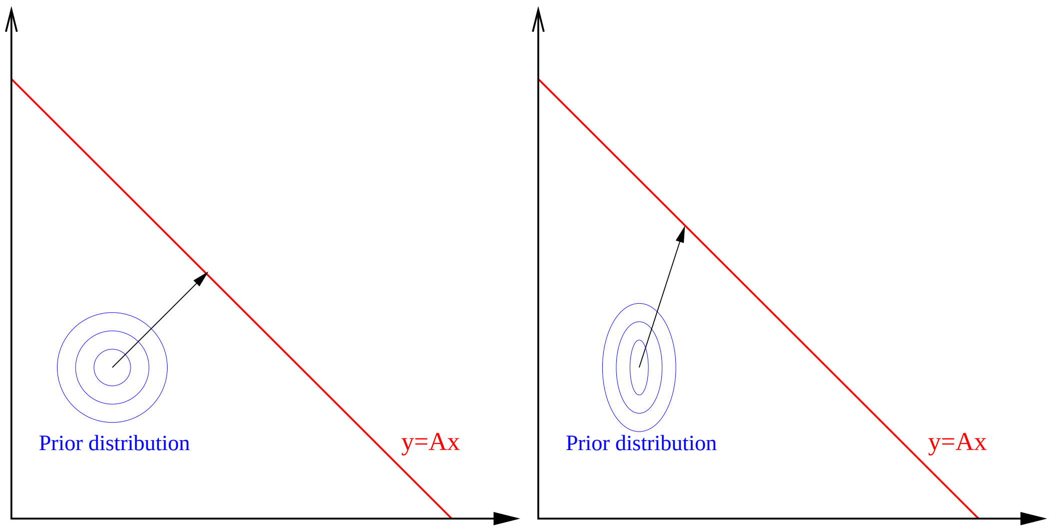


Figure 3.6: Effect of prior distributions covariance matrix

further to

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\lambda}_0), \tag{3.21}$$

which is just the shortest distance projection of the prior $\lambda_0$ to the subspace where the link count condition $\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{\lambda}$ is satisfied. We make use of this equation in our quick method in chapter 5.

If a covariance matrix is used, the effect is as depicted for the two-dimensional case in Figure 3.6. The covariance matrix has exactly the same function here as the weights had in tomogravity method of section 3.2.5. It can be understood intuitively so that as the variance in the figure is smaller in the direction of the horizontal axis than in the vertical direction, this means that we are more certain about the accuracy of the prior value for the horizontal co-ordinate axes, while having a larger uncertainty about the other variable. Thus, it is natural that the mean of the posterior is not far from the prior in the horizontal direction.

## 3.4 Maximum Likelihood Estimation

The extra information in the methods studied thus far has come from the gravity model priors. In maximum likelihood estimation (MLE) the source of the extra information is the second moment of the OD pair traffic volumes. By the use of the mean-variance relation, this can be used to help estimate the mean of the OD pair traffic. This approach requires that we have several consecutive link count measurements available, and that these are independently and identically distributed.

The size of the problem in traffic matrix estimation is typically quite large. Thus, numerical methods have to be used to find the likelihood estimate. The EM algorithm [33] is usually used, and we review its use in this context in section 3.4.4. However, the shape of the likelihood surface makes even the EM algorithm extremely slow to converge. Therefore, specific methods to make the MLE approach scalable to larger networks have been proposed. These are reviewed in sections 3.4.6 and 3.4.7.

Since the second moment plays such an integral part in MLE, we first introduce the second moment equation in section 3.4.1 and then move on to the methods.

### 3.4.1 Second Moment Equation

Again we have only the link count measurements available. Denote the covariance matrix of the link counts by $\boldsymbol{\Sigma}^{(y)}$, and $\boldsymbol{S}^{(y)}$ is a $\frac{1}{2}m(m+1)$-vector containing diagonal and upper triangle elements of the sample covariance matrix of the link counts. The covariance matrix of the OD pair counts, assumed to be a diagonal matrix, is similarly denoted in vector form by $\boldsymbol{S}^{(x)}$. The diagonality of the matrix means that the OD pairs are assumed to be independent. We studied this in section 2.2.3 and found this assumption not completely accurate, yet reasonably good.

The second moment equation is

$$\boldsymbol{S}^{(y)} = \boldsymbol{B}\boldsymbol{S}^{(x)}, \tag{3.22}$$

where $\boldsymbol{B}$ is a matrix containing the componentwise products of the rows of the routing matrix $\boldsymbol{A}$.

A row of $\boldsymbol{B}$ is labelled by a compound index $(ij)$, where $i = 1, \dots, m;\ j = i, \dots, m$, meaning that the index runs through $\frac{1}{2}m(m+1)$ values,

$$B_{(ij),k} = A_{i,k}A_{j,k} \qquad i = 1, \dots, m;\ j = i, \dots, m;\ k = 1, \dots, n$$

In vector form this can be written as

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{A}_1 \star \boldsymbol{A}_1 \\ \boldsymbol{A}_1 \star \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_1 \star \boldsymbol{A}_m \\ \boldsymbol{A}_2 \star \boldsymbol{A}_2 \\ \boldsymbol{A}_2 \star \boldsymbol{A}_3 \\ \vdots \\ \boldsymbol{A}_m \star \boldsymbol{A}_m \end{pmatrix},$$

where $\boldsymbol{A}_i$ denotes the $i$th row of $\boldsymbol{A}$, and the componentwise product is denoted with the star ($\star$). Note that a componentwise product of a row with itself is just that row. The rows of $\boldsymbol{B}$ indicate the elements of $\boldsymbol{x}$ contributing to covariance between links $i$ and $j$.

Apply this to our example network from chapter 1, recalling that the routing matrix in that case was

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

The number of rows in $\boldsymbol{A}$ is $m = 2$, thus there are

$$\frac{1}{2}m(m+1) = \frac{1}{2} \cdot 2 \cdot 3 = 3$$

rows in the matrix $\boldsymbol{B}$, which is

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{A}_1 \star \boldsymbol{A}_1 \\ \boldsymbol{A}_1 \star \boldsymbol{A}_2 \\ \boldsymbol{A}_2 \star \boldsymbol{A}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

The first row indicates the OD pairs using link $1$, thus contributing to the variance of the link. Analogously, the third row indicates the OD pairs contributing to the variance of the second link. The second row of the matrix indicates OD pairs using both links $1$ and $2$, contributing to the covariance between those links.

### 3.4.2 Network Tomography

The work by Vardi in [27] is one of the first papers on traffic matrix estimation in computer networks. Vardi was the first to propose a method using the second moments to serve as the additional information to make the system identifiable, and coined the term network tomography, because of the similarities in the problem to medical tomography.

A Poisson distribution is assumed, meaning that variance is equal to the mean

$$\boldsymbol{S}^{(x)} = \boldsymbol{\lambda}.$$

This allows the sample covariance matrix of the link counts to be used to estimate the traffic matrix. Basically, the method uses the mean-variance relation (2.1) with parameters $\phi = 1, c = 1$, according to the Poisson assumption.

Recall that the traffic matrix and the sample mean of the link counts are connected through the first moment equation (1.4)

$$\overline{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{\lambda}.$$

The second moment equation is now

$$\boldsymbol{S}^{(y)} = \boldsymbol{B}\boldsymbol{\lambda}.$$

These can be now combined, and the whole system becomes

$$\begin{pmatrix} \overline{\boldsymbol{y}} \\ \epsilon \boldsymbol{S}^{(y)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} \\ \epsilon \boldsymbol{B} \end{pmatrix} \boldsymbol{\lambda}. \tag{3.23}$$

Coefficient $\epsilon \in (0, 1]$ defines how much weight is given to the second moment estimate in the final solution, with $\epsilon = 1$ implying a strong faith in the Poisson assumption, while the smaller $\epsilon$ gets the more is the estimate based on the first moments.

This is a linear inverse positive, or LININPOS, problem and can be solved by numerical likelihood methods, such as the EM algorithm. The solution obtained this way minimizes the Kullback-Leibler distance between the observed moments and theoretical values.

Vardi's method does not give very accurate estimates, as was discovered by Gunnar et al. [12]. This is due to the fact that the Poisson assumption is not accurate in current IP networks. Thus, the general mean-variance relation and Gaussian distribution have to be used. In the next section the likelihood function under these assumptions is introduced.

### 3.4.3 Likelihood Function

In order to employ the likelihood approach, an assumption about the underlying distribution has to be made. For instance, we may assume that OD-pair traffic follows Gaussian distribution. If the distribution of the traffic matrix is

$$\boldsymbol{X} \sim N(\boldsymbol{\lambda}, \boldsymbol{\Sigma}) \tag{3.24}$$

it follows from (1.4) that the distribution of the link counts is

$$\boldsymbol{Y} \sim N(\boldsymbol{A\lambda}, \boldsymbol{A\Sigma A}^{\mathrm{T}}) \tag{3.25}$$

The likelihood function for the $\boldsymbol{\lambda}$, conditioned on $\tau$ measurements of $\boldsymbol{y}$, is now

$$L(\boldsymbol{\lambda}, \boldsymbol{\Sigma}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\tau) = \frac{1}{|\boldsymbol{A\Sigma A}^{\mathrm{T}}|^{\frac{\tau}{2}}} e^{-\frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{y}_t - \boldsymbol{A\lambda})^{\mathrm{T}}\left(\boldsymbol{A\Sigma A}^{\mathrm{T}}\right)^{-1}(\boldsymbol{y}_t - \boldsymbol{A\lambda})} \tag{3.26}$$

A computationally more convenient log-likelihood is

$$l(\boldsymbol{\lambda}, \boldsymbol{\Sigma}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\tau) = -\frac{\tau}{2}\log|\boldsymbol{A\Sigma A}^{\mathrm{T}}| - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{y}_t - \boldsymbol{A\lambda})^{\mathrm{T}}(\boldsymbol{A\Sigma A}^{\mathrm{T}})^{-1}(\boldsymbol{y}_t - \boldsymbol{A\lambda}). \tag{3.27}$$

The log-likelihood surface is depicted in Figure 3.7. The variables here are OD pairs $1$ and $3$ from the two link toy topology from Figure 1.1 of chapter 1. In the figure we have scaled the variables so that for each OD pair the true value is denoted by $10$. As we can see from the topology, the OD pairs in question share a link. Thus, the likelihood surface has a constant value ridge going diagonally through it from point $(0, 20)$ to point $(20, 0)$. That is to say, if $\lambda_1$ is increased and $\lambda_3$ decreased by the same value, the likelihood remains the same. Indeed, the ridge is exactly the line shown in Figure 1.2, which is the subspace satisfying the link count constraints. The existence of this constant value ridge follows directly from the fact that the system is not identifiable without some extra information.

In previous sections we have seen methods that start with a prior and then either project it to a point on the top of the ridge, or find some kind of compromise between the prior estimate and the ridge. In maximum likelihood estimation, we cannot have an ambiguous maximum for the likelihood surface. Thus, a relation between mean and variance is needed to alter the surface so that it has an unambiguous maximum point, making it possible to obtain accurate results. Imposing the mean-variance relation 2.1, the distribution becomes

$$\boldsymbol{Y} \sim N(\boldsymbol{A\lambda}, \phi\boldsymbol{A\Lambda}^{c}\boldsymbol{A}^{\mathrm{T}}) \tag{3.28}$$

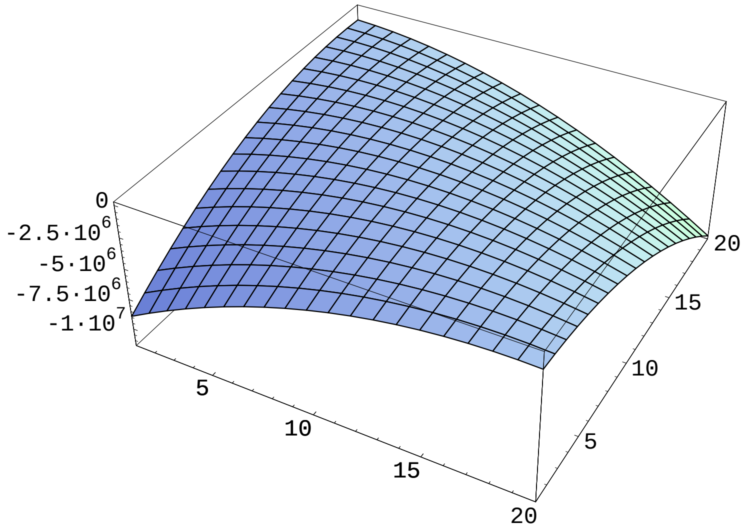


Figure 3.7: Likelihood surface without mean-variance relation

The log-likelihood function of the parameters $(\boldsymbol{\lambda}, \phi, c)$ conditioned on $\tau$ measurements of $\boldsymbol{y}$ is

$$
\begin{aligned}
&l(\boldsymbol{\lambda}, \phi, c|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\tau) = \\
&\quad -\frac{\tau}{2}\log|\phi \boldsymbol{A}\boldsymbol{\Lambda}^c\boldsymbol{A}^{\mathrm{T}}| - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\lambda})^{\mathrm{T}}(\phi\boldsymbol{A}\boldsymbol{\Lambda}^c\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\lambda}). \qquad (3.29)
\end{aligned}
$$

The new likelihood surface is depicted in Figure 3.8. Most notable change is that the cases where both variables would have small values are now extremely unlikely. The important thing is, however, that the ridge is no longer of constant height. Although there is still a clear ridge-like behavior, the mean-variance relation has changed the surface so that there is an unambiguous maximum point. This can be best seen from Figure 3.9, where only the values from this diagonal ridge are depicted. On the left side, the situation without mean-variance relation shows that the ridge is constant from corner to corner. On the right side, on the other hand, we have the new situation, where a maximum point is clearly observable.

Usually the size of the problem in traffic matrix estimation is so large, that numerical algorithms have to be used to find the maximum likelihood estimate. Most commonly the EM algorithm [33], which we present in the next section, is used.

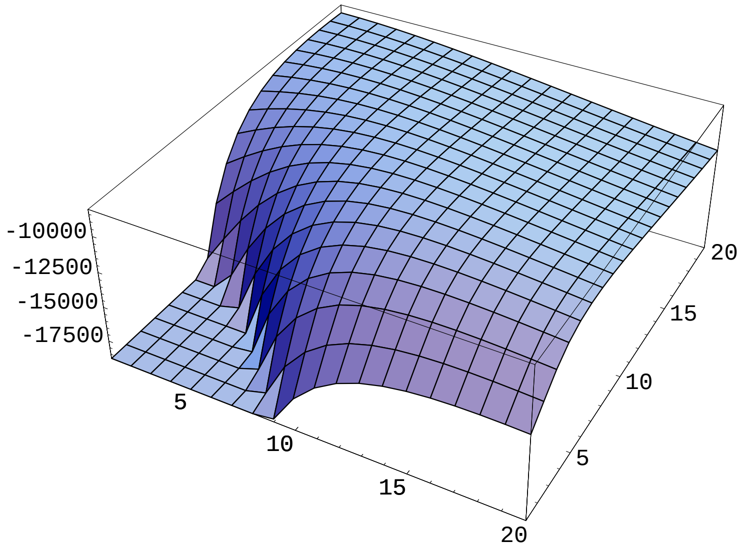


Figure 3.8: Likelihood surface with mean-variance relation

### 3.4.4 Expectation Maximization Algorithm

The Expectation Maximization (EM) [33] algorithm is a numerical method to solve maximum likelihood problems. In this section we give an overview of the use of the EM algorithm in traffic matrix estimation. See also [29] for a review on the subject.

We follow the formulations of Cao et al. [5], where the exponent parameter $c$ of the mean-variance power law is assumed to be constant. Thus, the parameters of the model are

$$\boldsymbol{\theta} = (\phi, \boldsymbol{\lambda}).$$

Now the problem can be solved numerically.

#### Formulation of the EM Algorithm

The EM algorithm is typically used in problems with missing data, and is thus applicable to the traffic matrix estimation problem, where the full data $\boldsymbol{x}$ is not observed, but a smaller set of linear combinations of $\boldsymbol{x}$ in the form of link count measurements $\boldsymbol{y}$. A parametric distribution needs to be assumed for the missing data, for example Poisson or Gaussian distribution, and the problem is then to estimate the parameters based on the observed data only. If the missing data were

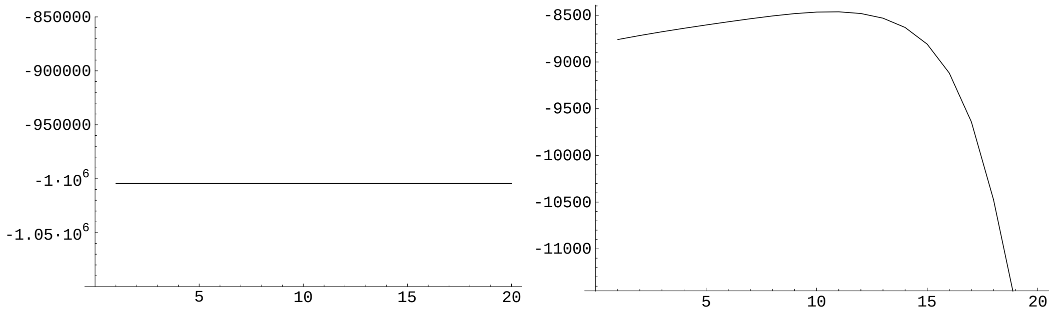


Figure 3.9: The constant value ridge disappears when the the mean-variance relation is imposed.

known, we could use the conventional likelihood estimation. That likelihood function is called the complete data likelihood in this framework. For our problem, the complete data log-likelihood is

$$l(\boldsymbol{\theta}|\boldsymbol{X}) = -\frac{\tau}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{x}_t - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t - \boldsymbol{\lambda}). \tag{3.30}$$

But since the $\boldsymbol{x}$ is missing, it is not possible to estimate $\boldsymbol{\theta}$ from this. The way this is handled in EM is by introducing an intermediate function $Q$, called the EM-equation. It gives the expected value of equation (3.30), given $\boldsymbol{y}$ and a previous value for the parameters $\boldsymbol{\theta}$.

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) \quad = \quad \mathrm{E}[l(\boldsymbol{\theta}|\boldsymbol{X})|\boldsymbol{Y}, \boldsymbol{\theta}^{(k)}],$$

where $\boldsymbol{\theta}$ is a variable and $\boldsymbol{\theta}^{(k)}$ the prior value for it. The parameters are updated iteratively, using the $Q$-function. With each iteration round defined as

$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\theta} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}).$$

The iteration has two steps. The E-step computes $Q$ function, that is, the distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ and $\boldsymbol{\theta}^{(k)}$. The second step is the M-step, where the $Q$ function is maximized, and the parameter value yielding the maximum is the value for $\boldsymbol{\theta}^{(k+1)}$.

In the traffic matrix estimation case, The $Q$ function is

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) \quad = \quad \mathrm{E}[-\frac{\tau}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{x}_t - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t - \boldsymbol{\lambda})|\boldsymbol{Y}, \boldsymbol{\theta}^{(k)}]$$

Since

$$\mathrm{E}[(\boldsymbol{x} - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\lambda})] \quad = \quad \mathrm{E}[\mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\lambda})(\boldsymbol{x} - \boldsymbol{\lambda})^{\mathrm{T}}\}]$$

$$
\begin{aligned}
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}\mathrm{E}[(\boldsymbol{x}-\boldsymbol{\lambda})(\boldsymbol{x}-\boldsymbol{\lambda})^{\mathrm{T}}]\} \\
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}\mathrm{E}[((\boldsymbol{x}-\boldsymbol{m})+(\boldsymbol{m}-\boldsymbol{\lambda}))((\boldsymbol{x}-\boldsymbol{m})+(\boldsymbol{m}-\boldsymbol{\lambda}))^{\mathrm{T}}]\} \\
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{R}+(\boldsymbol{m}-\boldsymbol{\lambda})(\boldsymbol{m}-\boldsymbol{\lambda})^{\mathrm{T}})\} \\
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}\boldsymbol{R}\}+(\boldsymbol{m}-\boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}-\boldsymbol{\lambda})
\end{aligned}
$$

we can write the $Q$-function in the form

$$
Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(k)}) = -\frac{\tau}{2}(\log|\boldsymbol{\Sigma}|+\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{R}^{(k)})) - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{m}_t^{(k)}-\boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}_t^{(k)}-\boldsymbol{\lambda}), \tag{3.31}
$$

where

$$
\begin{aligned}
\boldsymbol{m}_t^{(k)} &= \mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t,\boldsymbol{\theta}^{(k)}] \\
&= \boldsymbol{\lambda}^{(k)}+\boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y}_t-\boldsymbol{A}\boldsymbol{\lambda}^{(k)}), \qquad (3.32)\\
\boldsymbol{R}^{(k)} &= \mathrm{Var}[\boldsymbol{x}_t|\boldsymbol{y}_t,\boldsymbol{\theta}^{(k)}] \\
&= \boldsymbol{\Sigma}^{(k)}-\boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}^{(k)}. \qquad (3.33)
\end{aligned}
$$

The derivation of these equations is given in appendix A.

For updating the $\boldsymbol{\theta}$ the one-step Newton-Raphson algorithm is used

$$
\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - [\dot{\boldsymbol{F}}(\boldsymbol{\theta})]^{-1}\boldsymbol{f}(\boldsymbol{\theta}^{(k)}), \tag{3.34}
$$

where $\boldsymbol{f} = \partial Q/\partial\boldsymbol{\theta}$ and $\dot{\boldsymbol{F}}$ is the Jacobian of $\boldsymbol{f}$ with respect to $\boldsymbol{\theta}$.

### Convergence of EM

In one iteration of the EM, the algorithm yields a new estimate for the parameter vector based on the current estimate. That is, the value is updated

$$
\boldsymbol{\theta}^{(k)} \to \boldsymbol{\theta}^{(k+1)}.
$$

We can write this as an implicit mapping from the parameter space to itself, where one iteration step is defined by

$$
\boldsymbol{\theta}^{(k+1)} = M(\boldsymbol{\theta}^{(k)}).
$$

If the algorithm converges to the maximum likelihood estimate $\boldsymbol{\theta}^*$ it must satisfy

$$
\boldsymbol{\theta}^* = M(\boldsymbol{\theta}^*).
$$

The matrix $\boldsymbol{J}$ is defined as the Jacobian matrix of $M(\boldsymbol{\theta})$. Its element $(i,j)$ is thus

$$J_{ij}(\boldsymbol{\theta}) = \frac{\partial M_i(\boldsymbol{\theta})}{\partial \theta_j}.$$

This is called the matrix rate of convergence of the algorithm. We can approximate the elements of the matrix by

$$J_{ij}(\boldsymbol{\theta}) = \lim_{\epsilon \to 0} \frac{M_i(\boldsymbol{\theta} + \boldsymbol{\epsilon}_j) - M_i(\boldsymbol{\theta} - \boldsymbol{\epsilon}_j)}{2\epsilon}.$$

The largest eigenvalue $\lambda$ of the matrix $J(\boldsymbol{\theta}^*)$ is the global rate of convergence $r$. The global speed of convergence $s$ is one minus the rate of convergence, and is intuitively more appealing quantity, since a value close to zero indicates that the convergence is low, while a value close to $1$ would indicate fast convergence.

For example, in the case of the six node topology of Figure 5.2 that we use for simulations in chapter 5, the global speed of convergence obtained as

$$s = 1 - r = 0.00148. \tag{3.35}$$

This is an exceptionally low speed of convergence for the EM algorithm. In many typical problems the value of $s$ is between $0.1$ and $0.5$. The reason for the slow covergence is the hugely underdetermined nature of the problem, and the ridge like behavior of the likelihood surfaces. In our example it took up to $1000$ iterations to achieve convergences, while in many other problems of missing information ten iterations is sufficient.

**A Fast EM algorithm**

In their Technical report [18] Medina et al. propose a faster variant of the EM algorithm for traffic matrix estimation. They use additional linear constraints to add the rank of the routing matrix and then transform it to reduced echelon form. The key change is to abandon the Newton-Raphson approach of equation (3.34) in the M-step. Instead they transform the problem to solving a nonlinear equation by numerical methods by assigning

$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}). \tag{3.36}$$

This solution is obtained by solving a set of nonlinear equations using a procedure based on least square estimates that uses a trust region method and an interior-reflective Newton method. According to the authors this approach speeds the EM algorithm a good deal.

### 3.4.5 Time Varying Network Tomography

Using the maximum likelihood framework presented in the previous section, Cao et al. [5] formulate a time varying method for network tomography. They take a short window of measurements and use the EM algorithm to find the maximum likelihood estimate of the traffic matrix $\boldsymbol{\lambda}$ for that time slot. As the convergence of the EM algorithm is very slow, the authors propose to switch to a second order method based on quadratic approximation of the likelihood surface, when the change in EM iteration gets small.

The window is chosen small enough so that local stationary is a reasonable assumption, and the measurement sample is treated as independently and identically distributed random variables. Window size of $11$ is used, but $5$ and $21$ are reported to yield similar results. Thus, for time $t$

$$\boldsymbol{y}_{t-h}, \ldots, \boldsymbol{y}_{t+h} \sim iid\ N(\boldsymbol{A}\boldsymbol{\lambda}_t, \boldsymbol{A}\phi\boldsymbol{\Lambda}_t^c\boldsymbol{A}^{\mathrm{T}}),$$

where $h$ is the half-width of the window, in this case $h = 5$. The window is moved one time slot at a time, so that consecutive windows overlap in all but one measurement, which causes the estimated to be implicitly smoothed.

Once a time series of estimates of $\boldsymbol{\lambda}_t$ and $\phi_t$ are obtained, additional smoothing is performed on those values. Denoting

$$\boldsymbol{\eta}_t = (\log(\boldsymbol{\lambda}_t), \log(\phi_t))$$

the variable $\boldsymbol{\eta}$ is modelled as a random walk

$$\boldsymbol{\eta}_t = \boldsymbol{\eta}_{t-1} + \boldsymbol{v}_t \qquad \boldsymbol{v}_t \sim N(\boldsymbol{0}, \boldsymbol{V}),$$

where $\boldsymbol{V}$ is a variance matrix to be fixed beforehand, determining how much information carries over from preceding observations to the next. Letting $\tilde{\boldsymbol{Y}}_t$ denote the observation up to time $t$, a likelihood function is formulated for $\boldsymbol{\eta}_t$ as

$$p(\boldsymbol{\eta}_t|\tilde{\boldsymbol{Y}}_t) \sim p(\boldsymbol{\eta}_t|\tilde{\boldsymbol{Y}}_{t-1})p(\boldsymbol{Y}_t|\boldsymbol{\eta}_t). \tag{3.37}$$

The first term on the right hand side of equation (3.37) is approximated by first approximating $p(\boldsymbol{\eta}_t|\tilde{\boldsymbol{Y}}_{t-1})$ by a normal distribution

$$p(\boldsymbol{\eta}_{t-1}|\tilde{\boldsymbol{Y}}_{t-1}) \sim N(\hat{\boldsymbol{\eta}}_{t-1}, \hat{\boldsymbol{\Sigma}}_{t-1}),$$

where the posterior mode is used for mean, and the inverse curvature of the log posterior is used for the covariance matrix. Then

$$\pi(\boldsymbol{\eta}_t) = p(\boldsymbol{\eta}_t|\tilde{\boldsymbol{Y}}_{t-1}) \sim N(\hat{\boldsymbol{\eta}}_{t-1}, \hat{\boldsymbol{\Sigma}}_{t-1} + \boldsymbol{V}).$$

Now the log-likelihood equivalent of equation (3.37) becomes

$$g(\boldsymbol{\eta}_t) = \log \pi(\boldsymbol{\eta}_t) + \log p(\boldsymbol{Y}_t | \boldsymbol{\eta}_t).$$

The mode $\hat{\boldsymbol{\eta}}_t = \mathrm{argmax}\, g(\boldsymbol{\eta}_t)$ can now be found using numerical methods. Then let

$$\hat{\boldsymbol{\Sigma}}_t = \ddot{g}(\hat{\boldsymbol{\eta}}_t)^{-1}$$

and move to the next time slot $t+1$, and iterate until the whole time-series has been gone through, and a final estimate is obtained for $\boldsymbol{\lambda}_t$ for each $t$.

The problem with the time-varying network tomography is that it is rather time consuming. The authors conclude that it does not readily scale for realistic size networks. The problem has to be divided into smaller subproblems in order to handle larger networks.

### 3.4.6 Scalable Likelihood Approach

In [6] Cao et al. present a way to modify their method in [5] in order to use it for realistic size networks. They call the new method a divide-and-conquer approach, as the idea is to form several smaller subproblems and solve them separately.

All OD pairs are divided into disjoint sets $S_i$. For each set $S_i$ some subset $L$ of the links is chosen to estimate the OD pairs involved. The OD pairs that do not belong to the subset $S_i$, but are still using some of the selected links need to be involved in the estimation, but they can be aggregated together as much as possible. All the OD pairs using only a given link $j \in L$ can be all grouped together, for instance, as they are similar as far as this subproblem is concerned.

The most simple example of this approach is to formulate a separate subproblem for each OD pair. The simplest link set for this subset of one OD pair is just the first and last link in the path of that OD pair. As for the other OD pairs, all OD pairs using the first link are grouped together, all OD pairs using the last link are grouped together, and the rest are irrelevant for this subproblem. Now the familiar link count relation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ for this subproblem can be formulated as

$$\begin{pmatrix} y[s] \\ y[d] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x[sd] \\ x[s\overline{d}] \\ x[\overline{s}d] \end{pmatrix}$$

where $s$ is the source node and $d$ the destination node of the OD pair in question, while $\overline{s}/\overline{d}$ represents any other source/destination node.

The way to form subproblems larger than one OD pair is to select, for instance, set of nodes that are located near each other, or select OD pairs that are close to one another according to some distance parameter. Examples of distances used are

$$d(i,j) = \text{number of links used by } i \text{ or } j \text{ but not both},$$

which groups together OD pairs that a share a lot of links. A more complex distance criterion is

$$d(i,j) = 1 - \frac{|I_{ij}|}{\sqrt{I_{ii} I_{jj}}},$$

where

$$I_{ij}(\boldsymbol{\lambda}) = \langle \partial_i, \partial_j \rangle,$$

and $\partial_i$ is a information theoretic variable, namely the information direction corresponding to $\lambda_i$. That is to say, changing the distribution in this direction would change $\lambda_i$ the most of any direction. The disadvantages of this model are that the computation cost is larger than with the simpler approach. Also it depends on $\boldsymbol{\lambda}$ which is unknown.

Link selection for larger subproblem is carried out by first selecting the first and last link of each OD pair in the subset. Then, intermediate links and also the reversal of these links are added for each OD pair. At each intermediate node in the subnetwork, the incoming links not in $L$ are grouped together by forming a virtual superlink. Finally the OD pairs using exactly the same links from linkset $L$ are grouped together. The approach of [5], presented in section 3.4.5, can then be applied to each subproblem, yielding the estimates for traffic matrix elements.

The authors note that this scalable method works only when exponent value $c = 1$ is used in the mean-variance relation. This might be a big problem, since in many cases $c = 1$ is not a realistic parameter value.

### 3.4.7 Pseudo Likelihood Estimation

The Pseudo likelihood approach [15] is another scalable method for likelihood estimation. The idea is to use a slightly modified EM algorithm. The problem is divided into subproblems with each covering one OD-pair, as in the simple case example in the previous section. For each subproblem we denote the observed link counts on the subset of links used by $\boldsymbol{y}^s$, the unknown actual traffic counts of OD pairs involved by $\boldsymbol{x}^s$ and the sub-routing matrix comprised of the two rows corresponding to the two links by $\boldsymbol{A}^s$, while $\boldsymbol{\lambda}^s$ and $\boldsymbol{\Sigma}_s$ are the mean and covariance matrix of the traffic matrix respectively.

Since we have only two links per subproblem, the term $\boldsymbol{A}^s\boldsymbol{\Sigma}_s\boldsymbol{A}^{s\mathrm{T}}$ is only of dimension $2 \times 2$. Thus, there is no need to invert a large matrix, which makes the computations much faster. The complete data log likelihood of a subproblem $s$ is

$$l^s(\boldsymbol{\theta};\boldsymbol{X}^s) = -\frac{\tau}{2}\log|\boldsymbol{\Sigma}_s| - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{x}_t^s - \boldsymbol{\lambda}^s)^{\mathrm{T}}\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{x}_t^s - \boldsymbol{\lambda}^s).$$

In section 3.4.4 the Q-function for the EM algorithm was given in equation (3.31) as

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(k)}) \;=\; -\frac{\tau}{2}(\log|\boldsymbol{\Sigma}| + \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{R}^{(k)})) - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{m}_t^{(k)} - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}_t^{(k)} - \boldsymbol{\lambda}),$$

For the problem divided into subproblems it is

$$\begin{aligned} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(k)}) \;\propto\; \sum_{s\in S}\Bigg( & -\tau(\log|\boldsymbol{\Sigma}_s| + \mathrm{Tr}(\boldsymbol{\Sigma}_s^{-1}\boldsymbol{R}^{s(k)})) \\ & + \sum_{t=1}^{\tau}(\boldsymbol{m}_t^{s(k)} - \boldsymbol{\lambda}^s)^{\mathrm{T}}\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{m}_t^{s(k)} - \boldsymbol{\lambda}^s)\Bigg), \end{aligned}$$

where $m^s$ and $R^s$ are as given in section 3.4.4, only with the variables changed to corresponding subproblem variables.

The solution of this calls for the use of Multiple-step Gradient EM algorithm. The computational complexity of each EM step is now $O(n^{3.5})$ compared to the full likelihood method's $O(n^5)$. The authors report that for a small network studied the error in estimation accuracy increases only from $8\%$ to $9\%$ when switching from the full likelihood to the pseudo likelihood method.

## 3.5 Other Methods

This section reviews methods that use ways other than gravity modelling or second moment estimates to obtain the extra information for the problem. Linear programming methods try to substitute the extra information by a convenient choice of objective function, but we will argue in the next section that this approach does not work. The route change method increases the accuracy of a simple euclidian norm by changing the routing so that at different times different OD pairs are isolated so that they are observable, or at least more accurately estimated.

### 3.5.1 Linear Programming

Some efforts have been made to apply linear programming methods to traffic matrix estimation. The problem in this approach is the selection of a suitable object function.

Vaton et al. [29] note that a classical method to solve underdetermined linear systems is to minimize the euclidian norm

$$\begin{aligned} \min \quad & ||\boldsymbol{x}||^2 \\ \text{with} \quad & \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \end{aligned}$$

which has the solution

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^*\boldsymbol{y},$$

where $\boldsymbol{A}^*$ is the pseudo inverse of $\boldsymbol{A}$. They conclude that this is not a realistic approach for the traffic matrix estimation problem, as it finds the solution on the $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ subspace, that has the OD pairs as close to same size with each other as possible, which is not a realistic criterion.

Goldschmidt [11] formulates the problem with a weighted sum objective function

$$\max_{\boldsymbol{x}} \quad \sum_i w_i x_i \tag{3.38}$$

$$\text{subject to} \quad \sum_i A_{li} x_i \le y_l \qquad l = 1, \ldots, J \tag{3.39}$$

$$\sum_{l=(i,j)} y_l A_{lk} - \sum_{l=(j,i)} y_l A_{lk} = \begin{cases} x_k & \text{if } j = \text{source of } k \\ -x_k & \text{if } i = \text{destination of } k \\ 0 & \text{otherwise} \end{cases} \tag{3.40}$$

where (3.39) are the link count constraints and (3.40) are the flow conservation constraints. As we are trying to maximize (3.38), the method obviously tends to give as much bandwidth as possible to the OD pairs that are most beneficial to the objective function based on their weight function value and links used. Goldschmidt concludes that constant weights such as

$$w_i = 1 \qquad \forall i$$

do not work, but suggests that weights which are determined by the length of the path of each OD pair work quite well.

Consider again the two link example of Figure 1.1. The link counts in our example in chapter 1 were $y_1 = 10$, $y_2 = 9$. We showed that the region of feasible solutions is defined by

$$\begin{aligned} x_1 &= 10 - a, \\ x_2 &= 9 - a, \\ x_3 &= a \qquad a \in [0, 9]. \end{aligned}$$

The function to be maximized is thus

$$\begin{aligned} \sum_i w_i x_i &= w_1 x_1 + w_2 x_2 + w_3 x_3 \\ &= w_1(10 - a) + w_2(9 - a) + w_3 a \\ &= 10w_1 + 9w_2 - (w_1 + w_2 - w_3)a. \end{aligned}$$

Selecting constant weights $w_i = 1$ yields

$$\sum_i w_i x_i = 19 - a.$$

Since the only limitation is on link capacity, and adding to $x_3$ consumes the capacity of two links while contributing only to one OD pair, it is obvious that the maximum is obtained by choosing $a = 0$, which very likely is not the answer we are seeking.

In an attempt to overcome this problem, Goldschmidt proposed weights equalling the path length for each OD pair. The problem of traffic $x_3$ adding to the load of two links is countered by giving it twice the weight in the objective function. This yields

$$\begin{aligned} \sum_i w_i x_i &= 10 * 1 + 9 * 1 - (1 + 1 - 2)a \\ &= 19. \end{aligned}$$

Now the answer does not depend on $a$ at all! As long as we are on the feasible subspace defined by the link count constraints, the result is the same. So any point on the plane $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ gives the same value, as the whole feasible region is pareto-optimal. So while the real answer does yield the maximum value for the objective function with these weights, so would any other feasible answer. We have not gained any new knowledge about the situation by formulating it as a LP problem. By changing the weight from this equilibrium would instantly make some OD

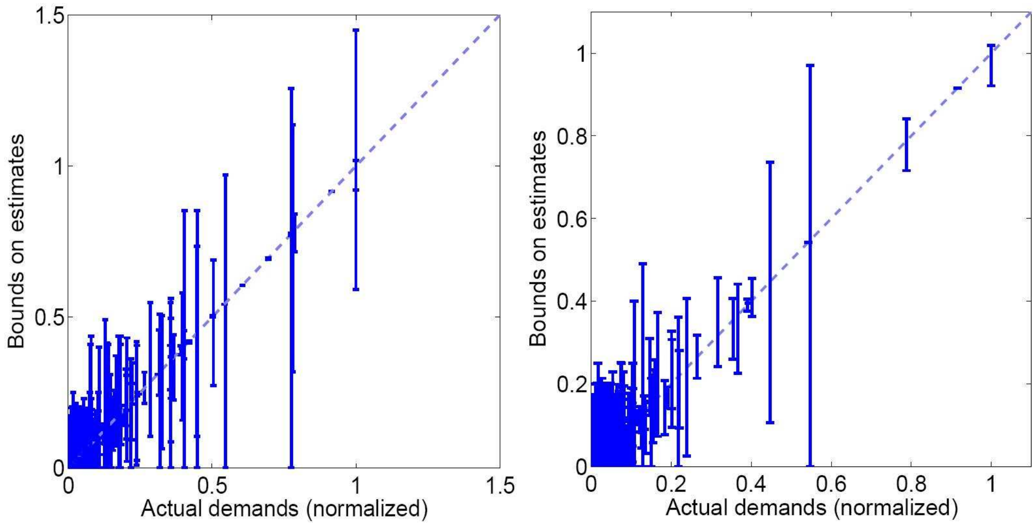


Figure 3.10: Examples of worst case bounds for Global Crossing's European (left) and North American (right) networks [12].

pairs more attractive to the objective function, and a maximum point would arise, but this would not be the answer we wish to find.

Medina et al. find in their study [16] that the method assigns zero values to many of the OD pairs. In [10] Eum at al. contribute the poor performance of Medina's study to their use of the simplex algorithm and demonstrate that the Interior point method yields accurate results. We can understand this by the fact that the whole feasible subspace, as mentioned above, is a pareto-optimal region. Thus, Simplex finds an extreme point from the boundary of the pareto-optimal region, while interior point method finds a random solution on the $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ subspace, depending on the initialization of the algorithm. However, as no extra information is brought in, there is no reason why this would be a better solution than any other random point in the middle of the feasible region.

A reasonable use for LP is given in [12], where the authors formulate worst case bounds for OD counts. That is, they use linear programming to find bounds for possible values of OD counts. The optimization problem is formulated as

$$\begin{array}{ll} \max(\min) & x_i \\ \text{subject to} & \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{array}$$

Obvious bounds are zero for lower bound and the lowest link count on the OD pair's path for upper bound, but in many cases it is possible to find tighter bounds for some OD pairs. The authors give examples from Global Crossing's European and North American backbone networks, shown in Figure 3.10, where the bounds for some OD pairs are surprisingly close to each other.

This method, however, is quite heavy computationally, as two LP problems need to be solved for each OD pair.

It is also possible to obtain a prior distribution from the worst case bounds by taking the average of lower bound and upper bound. This is shown in [12] to yield surprisingly adequate estimates.

### 3.5.2 Route Change Method

In [23], Soule et al. propose a method that achieves accurate results by changing routing. The idea is that in different routing scenarios different OD pairs are easier to estimate, as the links they use might be less heavily populated by other flows in some routing schemes than others. Each routing scheme is used for a period of $N$ measurements, and a total of $K$ different routings are used. $L$ denotes the number on links and $P$ the number of OD pairs.

The traffic is modelled as

$$\boldsymbol{x}(k,n) = \boldsymbol{\lambda} + \boldsymbol{w}(k,n)$$

where $k$ refers to the routing, and $n$ is a time index within that routing scheme. The link count equation is now

$$\boldsymbol{y}(k,n) = \boldsymbol{A}(k)\boldsymbol{x}(k,n) \qquad \forall k, \forall n. \tag{3.41}$$

As we have stated earlier, the problem is heavily underconstrained, and it is not possible to infer for $\boldsymbol{x}$ from the above equation. This is where the routing changes come in. The following matrices are defined to capture the situation of different routing matrices.

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}(0,0) \\ \vdots \\ \boldsymbol{y}(0,N-1) \\ \boldsymbol{y}(1,0) \\ \vdots \\ \boldsymbol{y}(K-1,N-1) \end{pmatrix} \qquad \boldsymbol{W} = \begin{pmatrix} \boldsymbol{w}(0,0) \\ \vdots \\ \boldsymbol{w}(0,N-1) \\ \boldsymbol{w}(1,0) \\ \vdots \\ \boldsymbol{w}(K-1,N-1) \end{pmatrix},$$

where $\boldsymbol{Y}$ is an $LKN$ -vector capturing the link counts of each measurement period and $\boldsymbol{W}$ is a $PKN$ -vector of the traffic fluctuations. The traffic matrices are collected in $\boldsymbol{A}$ as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}(0) \\ \vdots \\ \boldsymbol{A}(0) \\ \boldsymbol{A}(1) \\ \vdots \\ \boldsymbol{A}(K-1) \end{pmatrix},$$

which contains the routing matrices for each measurement period. Thus, there are $N$ copies of each routing matrix $\boldsymbol{A}(k)$. The dimensions of $\boldsymbol{A}(k)$ are $L \times P$, so $\boldsymbol{A}$ is an $LNK \times P$ matrix. And finally

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{A}(0) & 0 & \dots & 0 \\ 0 & \boldsymbol{A}(0) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{A}(K-1) \end{pmatrix}$$

is an $LNK \times PNK$ matrix.

Now (3.41) can be put into matrix notation

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\lambda} + \boldsymbol{C}\boldsymbol{W}. \tag{3.42}$$

Let $\boldsymbol{R}$ denote the covariance matrix of $\boldsymbol{W}$ and define

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}(0) & \boldsymbol{R}(1) & \dots & \boldsymbol{R}(T-1) \\ \boldsymbol{R}(1) & \boldsymbol{R}(0) & \dots & \boldsymbol{R}(T-2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{R}(T-1) & \boldsymbol{R}(T-2) & \dots & \boldsymbol{R}(0) \end{pmatrix},$$

where

$$\boldsymbol{R}(t) = [r_p(t)] = \mathrm{diag}(E[W_p(\tau)W_p(\tau+t)]).$$

As long as we assume that the covariance matrix is a diagonal matrix, meaning the OD pairs are uncorrelated, we can solve $\boldsymbol{R}(t)$ based only on link count covariance matrix. We show this in detail in section 5.2, as it is a key part of our quick method also. For now, it suffices to assume that $\boldsymbol{R}$ is known.

The minimum mean square error estimate for $\boldsymbol{\lambda}$ is now

$$\hat{\boldsymbol{\lambda}} = \left(\boldsymbol{A}^T(\boldsymbol{C}\boldsymbol{R}\boldsymbol{C}^T)^{-1}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T(\boldsymbol{C}\boldsymbol{R}\boldsymbol{C}^T)^{-1}\boldsymbol{C}\boldsymbol{W}. \tag{3.43}$$

This reduced to the pseudo-inverse solution when $\boldsymbol{R}$ is the identity matrix. The fact that $\boldsymbol{R}$ is known and that we have several different routings involved makes the method work. The extra information that is always needed in underconstrained problems comes from the additional routing scenarios.

For the test topology in [23] $24$ route changes were needed, to identify all OD pairs. Thus, it is obvious that the whole measuring period is so long that local stationarity will not hold. Therefore the diurnal variation has to be taken into account in the model. So insted of

$$\boldsymbol{x}(t) = \boldsymbol{\lambda} + \boldsymbol{w}(t),$$

the traffic has to modelled as

$$\boldsymbol{x}(t) = \boldsymbol{\lambda}(t) + \boldsymbol{w}(t),$$

where $\boldsymbol{\lambda}(t)$ is cyclo-stationary, with a $24$ hour period. This is done through the use of Fourier expansion

$$\boldsymbol{\lambda}(t) = \sum_{h=1}^{2N_b} \boldsymbol{\theta}_h b_h(t),$$

where for each $h$, $\boldsymbol{\theta}_h$ is a vector of coefficients and $b_h$ is a scalar basis function that is periodic with the same period as the traffic mean $\boldsymbol{\lambda}$. $\boldsymbol{x}(t)$ is represented as a weighted sum of $2N_b + 1$ basis functions. Vector $\boldsymbol{\theta}$ comprises of all the vectors $\boldsymbol{\theta}_h$, and the routing matrix is now written

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}'(0) \\ \boldsymbol{A}'(1) \\ \vdots \\ \boldsymbol{A}'(T-1) \end{pmatrix},$$

where

$$\boldsymbol{A}'(t) = \begin{pmatrix} \boldsymbol{A}(k)b_0(t) & \boldsymbol{A}(k)b_1(t) & \dots & \boldsymbol{A}(k)b_{2N_b}(t) \end{pmatrix}.$$

So instead of (3.42) the system is now written

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{C}\boldsymbol{W}. \tag{3.44}$$

This is essentially the same method, but takes into account the diurnal pattern and estimated $\boldsymbol{\theta}$, and hence $\boldsymbol{\lambda}$.

To reduce this quite large number of changes, the authors propose to pick out only the OD pairs with largest variance, trusting the mean-variance relation to the point that these are more or less also the OD pairs with largest means. If only the top $30\%$

of OD pairs, contributing $95\%$ of traffic, are estimated, the number of route changes needed is reduced to five.

The problem of designing the weight changes to optimally generate the different routings is tackled by the authors in [20].

## 3.6 Future Directions

Finally, we take a brief look into a new direction on the field of traffic matrix estimation. Two of the three methods reviewed in this section are proposed in [24], where the authors coin the term third generation methods to describe them. The common denominator for all these methods is that they need $24$ hours worth of direct measurements for calibration every now and then. So Netflow, or equivalent measuring, has to be available network-wide whenever needed. The classical Traffic matrix estimation framework rests on the assumption that this is not the case. Even if all routers would support Netflow, the overhead of measuring and transportation costs might be high. On the other hand, it is obvious that adding direct measurements improves vastly the accuracy of the estimates.

### 3.6.1 Fanout Method

The fanout method [21] does not use the routing matrix, but relies on measurements alone to obtain the traffic matrix. Let $f(i,j,t)$ denote the fraction of traffic entering node $i$ at time $t$ that is terminating at node $j$.

$$f(i,j,t) = \frac{x(i,j,t)}{\sum_j x(i,j,t)}.$$

Node $i$ has then a *baseline fanout*

$$\boldsymbol{f}(i,*,t) = \{f(i,j,t) \ \ \forall j\}.$$

The fanouts for each node are defined from the calibration measurements. As was discussed in section 3.2.4 and found also in [21], the fanouts are surprisingly stable. Thus the calibrated fanouts at time $t$ can be used to estimate the fanouts at the same time of day in subsequent days. The estimate for the traffic matrix is obtained from

$$\hat{x}(i,j,t) = \hat{f}(i,j,t)x(i,*,t),$$

where $\hat{f}$ is the estimated fanout and $x(i, *, t)$ is the total traffic incoming in node $i$, which can be obtained from SNMP measurements.

As the fanout do not remain constant forever, each node checks the validity of current fanouts by doing direct measurements for a $10$ minute period only. If the fanouts for this time slot have changed more than a preselected acceptable threshold, then a new calibration is performed.

### 3.6.2 Principal Components Method

In [14] it was found using principal component analysis (PCA) that when considering long time scales the OD flows can be captured by a lower dimensional representation of eigenflows. In fact, so much lower, that the number of required components is lower than the number of links. Therefore, while the traffic matrix inference is ill-posed problem, the aforementioned components can be estimated from the link counts, and thus the traffic matrix can be estimated as well [24].

Let $\boldsymbol{X}$ denote the time series of all OD pairs, so that its dimensions are $\tau \times n$. Using PCA this can be decomposed as

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}},$$

where $\boldsymbol{U}$ is a $\tau \times n$ matrix comprised of the eigenflow time series, $\boldsymbol{V}$ is $n \times n$ with principal components as its columns, and $\boldsymbol{S}$ is an $n \times n$ diagonal matrix where element $S(i, i)$ is a measure of energy captured by the principal components $i$. Calibration measurements are needed to formulate this presentation. Then it is possible to pick out the $k$ most important components, and the traffic matrix is approximated by

$$\boldsymbol{x}_t \approx \boldsymbol{V}'\boldsymbol{S}'\boldsymbol{u}'_t \qquad t = 1, \ldots, \tau, \tag{3.45}$$

where $\boldsymbol{V}'$ is a $n \times k$ matrix that includes the top principal components, $\boldsymbol{S}'$ is the corresponding diagonal matrix and $\boldsymbol{u}_t$ comprises of the $k$ most significant eigenflows.

Now the familiar link count equation has the form

$$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{V}'\boldsymbol{S}'\boldsymbol{u}'_t,$$

which is well-posed, and we can solve for $\boldsymbol{u}_t$ by taking the pseudo-inverse of $\boldsymbol{A}\boldsymbol{V}'\boldsymbol{S}'$. Then inserting $\hat{u}_t$ in (3.45) yields the estimate for the traffic matrix.

A re-calibration is performed when the theoretical link counts $\boldsymbol{A}\boldsymbol{x}_t$ calculated from the estimates differ from the observed link counts $\boldsymbol{y}_t$ more than a pre-selected acceptable threshold.

### 3.6.3 Kalman Filtering

In the Kalman Filtering method [24, 25] the traffic evolution is modelled according to the linear system

$$\boldsymbol{x}_{t+1} = \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{w}_T,$$

where $\boldsymbol{C}$ is a state transition matrix capturing the deterministic components of the traffic process, while $\boldsymbol{w}$ is a noise term. The diagonal terms of the transition matrix capture the temporal evolution of an OD pair, while the non-diagonal elements correspond to cross-correlation between OD pairs, should any exist. The link count equation is written here in the form

$$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{m}_t,$$

where $\boldsymbol{m}_t$ is measurement noise.

Letting $\hat{\boldsymbol{x}}_{t|t-1}$ be the prediction of $\boldsymbol{x}_t$ at time $t$ based on information up to time $t-1$, and $\hat{\boldsymbol{x}}_{t|t}$ is the estimation of $\boldsymbol{x}_t$ at time $t$ that adds the most recent measurement to the prediction. The task is to determine $\hat{\boldsymbol{x}}_{t+1|t+1}$. The prediction step is

$$\begin{aligned} \hat{\boldsymbol{x}}_{t+1|t} &= \boldsymbol{C}\hat{\boldsymbol{x}}_{t|t}, & (3.46) \\ \boldsymbol{P}_{t+1|t} &= \boldsymbol{C}\boldsymbol{P}_{t|t}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{Q}, & (3.47) \end{aligned}$$

where $\boldsymbol{P}$ is the covariance matrix of errors and $\boldsymbol{Q}$ is the covariance matrix of the noise term $\boldsymbol{W}$.

The estimation step uses the prediction as well as the measurement $\boldsymbol{Y}_{t+1}$ to update the state and variance:

$$\begin{aligned} \hat{\boldsymbol{x}}_{t+1|t+1} &= \hat{\boldsymbol{x}}_{t+1|t} + \boldsymbol{G}_{t+1}\left[\boldsymbol{y}_{t+1} - \boldsymbol{A}\hat{\boldsymbol{x}}_{t+1|t}\right], & (3.48) \\ \boldsymbol{P}_{t+1|t+1} &= \left(\boldsymbol{I} - \boldsymbol{G}_{t+1}\boldsymbol{A}\right)\boldsymbol{P}_{t+1|t}\left(\boldsymbol{I} - \boldsymbol{G}_{t+1}\boldsymbol{A}\right)^{\mathrm{T}} + \boldsymbol{G}_{t+1}\boldsymbol{R}\boldsymbol{G}_{t+1}, & (3.49) \end{aligned}$$

where $\boldsymbol{R}$ is the covariance matrix of measurement noise $\boldsymbol{M}$, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{G}$ is the *Kalman gain matrix*. Now equations (3.46) - (3.49) define the algorithm to obtain the linear minimum variance estimator, the Kalman filter, for the traffic matrix.

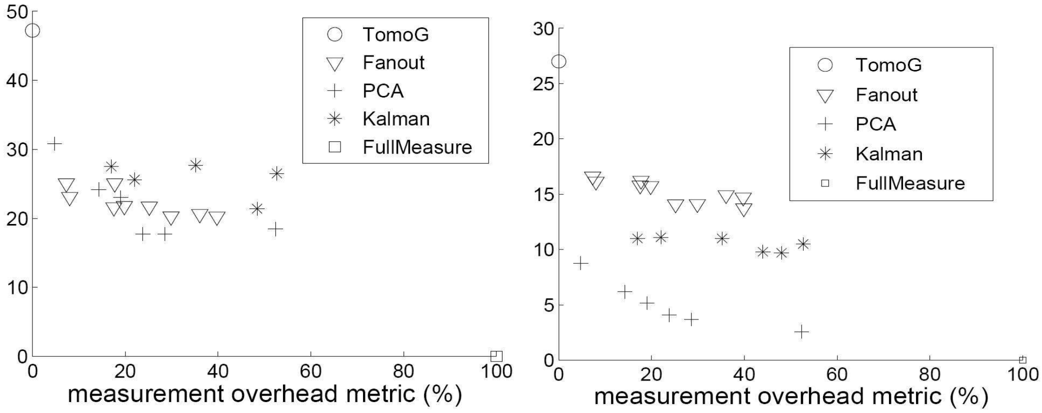


Figure 3.11: Error as a function of measurement overhead. Left: spatial errors, right: temporal errors [24].

Calibration measurements are needed to obtain $\boldsymbol{C}$,$\boldsymbol{Q}$ and $\boldsymbol{R}$ as well as initial values $\hat{\boldsymbol{x}}_{0|0}$ and $\boldsymbol{P}_{0|0}$. Re-calibration is performed as in the PCA method, when the theoretical link counts based on the estimates deviate too much from the observed link counts.

### 3.6.4 Tradeoff between Error and Overhead

The more measurements are made, the more accurate the estimates are. If direct measurements are made constantly, the error is zero but the overhead is very large. The classical traffic matrix estimation methods on the other hand, have zero overhead, but larger errors. The third generation methods each make a tradeoff between these two cases. In Figure 3.11 the error of the various methods is presented as a function of the measurement overhead. The scatter plot of a single method corresponds to different values of the re-calibration thresholds. The Tomogravity method is used to represent the classical methods. The spatial error on the left side of the figure calculates the error for each OD pairs over the whole time period, and then gives the average of these errors. The temporal error on the right side calculates the error for each time slot and gives the average of these over the considered time period. The error unit is the average relative L2 error percentage, and the measurement cost is scaled so that the overhead of full measurement is denoted by $100\%$.

It can be concluded that for stricter thresholds the Netflow is running half the time, or the calibration is done every other day on average. It is obvious that such massive measurement amounts yield accurate estimates. However, even with a measurement overhead of about one tenth of full measurement, the errors can be cut to half compared to the Tomogravity method, with the PCA method appearing to be the most effective. It must noted, however, that as we will see in section 4.2, the Tomogravity is not the most effective of the classical methods, but the MLE yields $25\%$ more accurate estimates on average.

# Chapter 4

# Evaluations of Estimation Methods

*In this chapter we give an overview of the comparisons of the accuracy of different methods found in the literature currently. Most papers give some simulation results or test cases, which justify why that particular proposed method makes sense. We are, however, only interested in objective comparisons made between different methods, preferably ones where none of the methods considered are by the authors themselves. Although there are already a few methods proposed as we saw in Table 3.1, there are not many comprehensive comparisons.*

## 4.1 Comparison by Medina et al. [16]

**Evaluated methods**

- Bayesian inference (section 3.3.1)
- Maximum Likelihood estimation (section 3.4.5)
- Linear Programming (section 3.5.1)

This is the first effort to compare the accuracy of different traffic matrix estimation methods.

First a study with a four node toy topology is conducted. The results reveal the breakdown of the LP method, discussed in section 3.5.1. For this reason it is not included in the comparison with a realistic size network.

The actual comparison is carried out using synthetic traffic simulations with a $14$-node backbone-like topology. The authors create several different synthetic traffic matrices with different OD pair distributions, including Gaussian, Poisson, uniform, constant and bimodal. The constant traffic matrix (TM) is obtained by assigning value $x_i = 300$ for each time period of each OD pair. The uniform TM is obtained by drawing

$$x_i \sim \mathrm{U}(200, 500) \qquad \forall i.$$

Poisson TM is obtained by first drawing the parameter values $\lambda_i$ from the uniform distribution

$$\lambda_i \sim \mathrm{U}(200, 500) \qquad \forall i,$$

and then the values for $x_i$ are drawn from Poisson distribution

$$x_i \sim \mathrm{Poisson}(\lambda_i) \qquad \forall i.$$

The means for the Gaussian case are drawn similarly as in the Poisson case, and the observed values are then drawn as

$$x_i \sim \mathrm{N}(\lambda_i, 40^2) \qquad \forall i.$$

This is an interesting choice, as the mean-variance relation assumption, which is the basis for the likelihood methods, is discarded from the start, instead using the same variance for each OD pair regardless of its mean. Most recent studies do indicate that there is at least a vague mean-variance relation prevailing in the traffic matrices of IP networks. For the bimodal case the TM is generated as a mixture of two Gaussians, with the value drawn from $\mathrm{N}(150, 20^2)$ with probability $0.8$, and from $\mathrm{N}(400, 20^2)$ with probability $0.2$.

The Bayesian method needs a prior distribution and even the EM algorithm needs a starting value. Thus, priors $\boldsymbol{x}'$ are generated by

$$x_i' = x_i + \epsilon,$$

where $\epsilon \sim \mathrm{N}(0, 60^2)$ for the so called "good prior" and $\epsilon \sim \mathrm{N}(0, 100^2)$ for the "bad prior." These are synthetic priors which are unbiased, since the expected value is the real value of the traffic matrix element, much like an estimate our Quick method without projection (section 5) would yield: unbiased, yet not very efficient. The often used gravity priors are, however, biased, even though they might be accurate.

As the Bayesian method considered uses only a one-sample snapshot of the link counts, the network tomography is used with window length of one measurement,

Table 4.1: Comparison results from Medina et al. [16]

| | Bayes(good prior) | EM(good prior) | Bayes(bad prior) | EM(bad prior) |
|---|---|---|---|---|
| Constant | 0.20 | 0.12 | 0.41 | 0.22 |
| Uniform | 0.26 | 0.13 | 0.43 | 0.24 |
| Poisson | 0.23 | 0.11 | 0.37 | 0.23 |
| Gaussian | 0.23 | 0.14 | 0.41 | 0.24 |
| Bimodal | 0.41 | 0.22 | 0.63 | 0.39 |

to make the comparison fair. This, of course, further hinders the use of the second moment estimates.

The results of the simulation study are given in Table 4.1. With the good prior, Bayesian method has an average error of $27\%$ while the network tomography by the EM algorithm has an average error of $14\%$. With the bad prior the average errors are $45\%$ and $26\%$ respectively. It seems that the likelihood approach is clearly better, even though the synthetic data did not have any relation between mean and variance. The Bayesian method of course is only as good as the prior used.

The most surprising result is that also the accuracy of the EM depends on the prior. In my opinion this should not be the case, since the extra information comes from the second moment estimate, not the prior. The shape of the likelihood surface certainly does not change if the prior is worse than before, so the maximum point of that surface is still the same. The only reason for the worse results might be that the algorithm converges to some local maximum. In [5], Cao et al. themselves state that the choice of the starting point is somewhat arbitrary in the EM algorithm.

Finally, it is concluded that longer measurement intervals yield only minor gains on the accuracy of the network tomogarphy method. This is understandable, but only under the synthetic traffic used here. As the mean-variance relation does not hold at all in their data, the fact that more measurements make the second moment estimate more accurate does not help in estimating the mean, as the more accurate variance estimator still gives no indication of the mean. If the mean-variance relation would hold, more samples would make the estimator more accurate.

## 4.2 Comparison by Medina et al. [18]

### Evaluated methods

- Tomogravity (section 3.2.5)
- Maximum likelihood estimation (section 3.4.5)

The synthetic data in this evaluation is constructed from real traffic measurements, in which, along with the link counts, the OD pair counts are available for three rows of the traffic matrix. This data is used to determine an empirical distribution for the fanouts, and the synthetic traffic model is constructed accordingly.

Maximum likelihood method obtained by the EM algorithm is compared with the tomogravity method. In Figure 4.1 the target traffic matrix is depicted by the almost solid line and the estimates by the scattered circles. It is clearly observable from the Figure that the MLE is more accurate in this situation. In the situation of Figure 4.2 the gravity prior is replaced by an unsuitable skewed prior distribution. This is also used for the starting point of the EM iteration. The tomogravity method suffers from the unsuitable prior distribution, as expected. The methods relying on priors to bring in the extra information, can be only as good as the prior allows. The MLE is not affected, as was also expected.

The authors do not provide tables of estimate errors, but state that the MLE approach is consistently about $25\%$ more accurate. In Figure 4.3 another kind of presentation

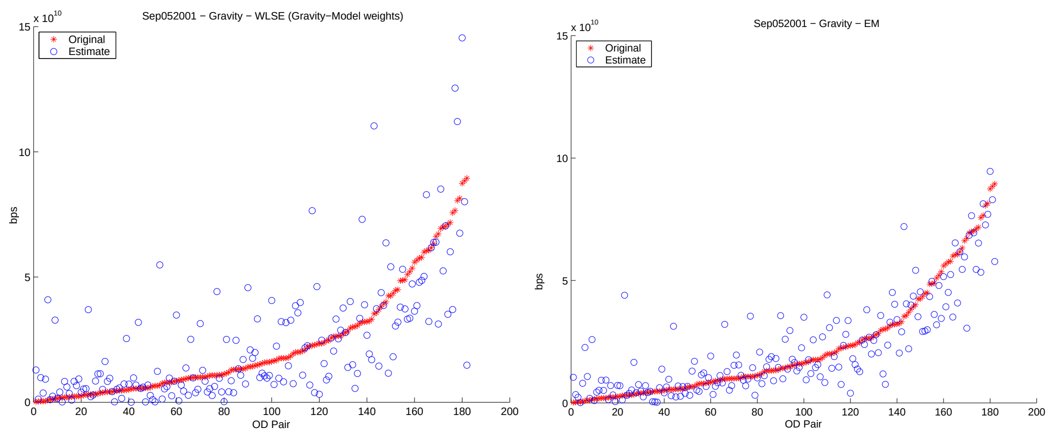


Figure 4.1: Estimates with Gravity prior. Left: Tomogravity. Right: MLE [18]

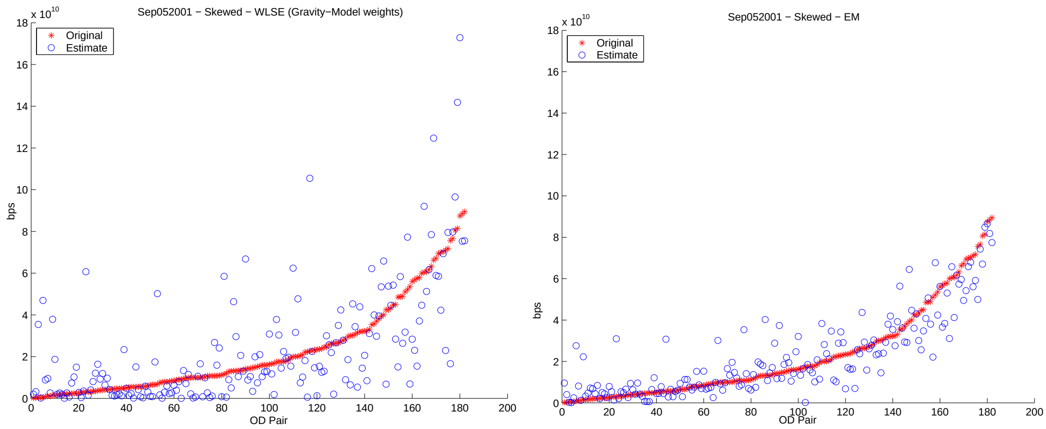


Figure 4.2: Estimates with skewed prior. Left: Tomogravity. Right: MLE [18]

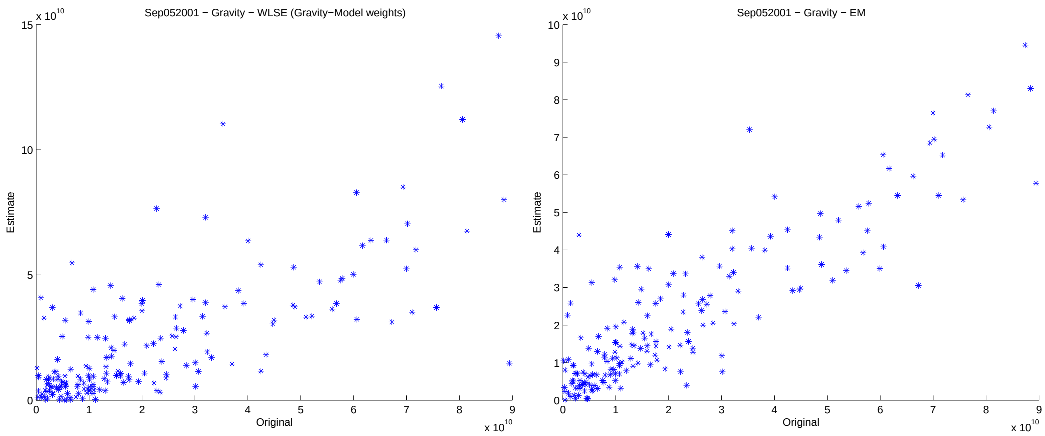


Figure 4.3: Left: Tomogravity. Right: MLE [18]

Table 4.2: Correlation coefficients for estimates with different priors

| prior | WLSE | MLE |
|---|---|---|
| Skewed | 0.1 | 0.8 |
| Choice | 0.2 | 0.8 |
| Gravity | 0.3 | 0.8 |

of the estimation accuracy is given, where the axis correspond to actual target value and the estimate yielded by the method. The closer to a line with $45$ degree slope the values are, the better the estimates. This accuracy can be quantified by the Pearson correlation coefficient $R$ between the plotted points and the line. These values are given in Table 4.2. The authors call the tomogravity method WLSE (Weighted least square estimation), since they use priors other than just gravity model. Inlcuding the unsuitable skewed prior and the Choice model (section 3.2.3). The best results are obtained by the traditional gravity model. Still the MLE is easily more accurate than the tomogravity method.

## 4.3 Comparison by Gunnar et al. [12]

**Evaluated methods**

- Bayesian inference (section 3.3.1)
- Information theoretic approach (section 3.2.6)
- Vardi's tomography (section 3.4.2)

This study is performed utilizing real measurements from Global Crossing's backbone networks. Due to MPLS capabilites in the network, the SNMP measurements provide the OD pair counts. A $250$ minute busy period, for which local stationary more or less holds, is considered. The performance of a method is quantified by the mean relative error of the largest OD pairs, where the largest OD pairs include those that together comprise about $90\%$ of total traffic in the network.

Vardi's Network tomography is found to give quite inaccurate estimates. Since the Poisson assumption does not hold in the studied network, regularization parameter

Table 4.3: Comparison results from Gunnar et al. [12]

| | Vardi | Entropy | Bayes |
|---|---|---|---|
| Europe | 0.47 | 0.11 | 0.08 |
| America | 0.98 | 0.22 | 0.25 |

value $\epsilon = 1$ yield results where some of the estimates are several orders of magnitude larger than the actual values. With $\epsilon = 0.01$ the values are much better but still not satisfactory. Mean relative errors with the sample of $50$ measurements is $0.47$ and $0.98$ for European and North American networks respectively. Unfortunately the authors do not include any method relying on the general mean-variance relation in their comparison.

The comparison between information theoretic approach, which the authors call Entropy method, and the Bayesian approach is done using a single measurement snapshot. The Gravity prior is used for the Bayesian method, as it is also the prior in the Entropy method. The Results are shown in Table 4.3. A marginal improvement is obtained for the Bayesian method by using the worst case bound prior proposed by Gunnar et al.

Either way, the result is a tie, so to speak, as the Bayesian method is slightly better for the European network and the Information theoretic approach for the North American network.

## 4.4 Comparison by Soule et al. [24]

**Evaluated methods**

- Tomogravity (section 3.2.5)
- Route Change method (section 3.5.2)

This study emphasizes the so-called third generation methods that were presented in section 3.6, where we discussed some of the results. However, in addition it does include some evaluations of Tomogravity and the Route change method. Since this is the only study to evaluate the latter, we conclude this chapter with a brief overview of the results.

The data used here are from Sprint's European backbone, from which three weeks worth of OD measurements were obtained in 2003 by enabling Netflow on all peering and access links.

The errors for Tomogravity method vary from $20\%$ to $33\%$, depending on the time slot considered. For the Route change method the errors are between $30\%$ and $50\%$. The authors demonstrate that the reason for the poor performance of the method is that it yields a perfectly cyclical pattern due to the Fourier transformation approach. The assumption that each OD pair is cyclo-stationary is so strong, that the model cannot capture any traffic characteristics deviating from this. Figure 4.4 shows an example of this behavior. The lower diagram depicts the estimate of the Route Change Method. It is clearly cyclo-stationary, thus failing to follow the traffic as well as the Tomogravity estimate, depicted in the upper diagram.

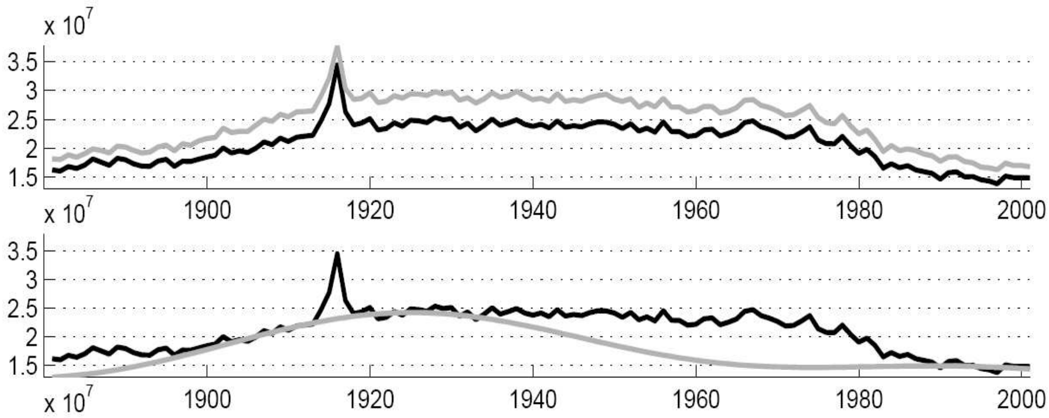


Figure 4.4: A short sample of actual traffic (black) against estimated traffic (grey).Top: Tomogravity, bottom: Route Change method[24]

# Chapter 5

# Quick Method Based on Link Covariances

*In this chapter we propose a novel method for traffic matrix estimation: The Quick method based on link covariances. We presented this method first in [3], and it is the main original contribution of the thesis.*
*As stated earlier some additional information other than just the link loads has to be brought in to obtain a solution for the estimation problem. In this method we use the link count sample covariance matrix. We propose two computationally light-weight methods based on the covariance matrix, the projection method and constrained minimization method. The accuracy of these methods is compared with that of other methods using second moment estimates by simulation under synthetic traffic scenarios.*

## 5.1 Introduction

Typically, methods that need a prior distribution use the gravity model to obtain one. However, the gravity assumption that the traffic volume of an OD pair is proportional to the total traffic sent by the origin node and the total traffic terminating at the destination node, does not always hold. In [12] the authors study real traffic matrix of a North American backbone network and conclude that there are significant errors concerning the estimation of the largest OD pairs, which are the most important ones for traffic engineering purposes. Therefore, we propose another way of obtaining prior distribution based on the link count covariances and the functional mean-variance relationship. Based on this, we develop two computationally light-weight methods, similar in principle to the tomogravity method of [30], in the sense that they incorporate a prior distribution and link count measurements to obtain an estimate.

The approximate relationship between link count sample mean and the traffic matrix $\boldsymbol{\lambda}$ was given in equation (1.4) of chapter 1 as

$$\overline{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{\lambda}. \tag{5.1}$$

We recall from chapter 3.4 that the Maximum likelihood method relies on the fact that the system of first and second order link count statistics together make the system identifiable with regard to the first order OD-pair statistic, using the mean-variance relation.

But, in fact, the second order statistic for OD-pairs is identifiable based solely on the second order statistic of the link counts, as long as we assume independence among OD-pairs and a sensible routing scheme. This result is proven by Soule et al. [23]. Since we can analytically solve the variance of the OD-pairs by least squares method, and the power-law relation between variance and mean is assumed, we can solve the traffic matrix from our variance estimate.

The benefit is that this does not call for numerical methods, and is thus extremely quick to calculate. The problem with this approach is that it does not take into account the link count equation (1.4), which is a stronger condition as opposed to the mean-variance relation which is only an assumption. Therefore, we propose two simple methods that incorporates this information into the solution obtained through estimation of the variance, yet maintaining the computational simplicity of the model.

## 5.2 Solving OD-pair Covariance Matrix from Link Counts

The MLE relies on the fact that the system of first and second order link count statistics together make the system identifiable with regard to the first order OD-pair statistics, i.e. we are able to find the solution for the likelihood equations if there exists a functional relationship between the mean and the variance of OD-pair traffic. As presented in previous chapters, the commonly used relation is the power-law relation

$$\mathbf{\Sigma} = \phi \cdot \mathrm{diag}\{\boldsymbol{\lambda}^c\}, \tag{5.2}$$

where $\mathbf{\Sigma}$ is a diagonal matrix, because we assume independence between OD pairs. Let us denote the number of links by $m$ and the number of OD-pairs by $n$. Then the vector form of traffic matrix $\boldsymbol{x}$ has the dimension $(n \times 1)$ and link loads $\boldsymbol{y}$ has the dimension $(m \times 1)$.

In section 3.4.1 the second moment equation of the traffic volume was defined in (3.22) as

$$\boldsymbol{S}^{(y)} = \boldsymbol{B}\boldsymbol{S}^{(x)}. \tag{5.3}$$

In Vardi's approach (3.23) the Poisson assumption makes it possible to just replace $\boldsymbol{S}^{(x)}$ with $\boldsymbol{\lambda}$ and write the first and second moment equations as a single formula. Now we have the general power-law relation instead of the Poisson assumption, so we cannot do that. Thus, we first solve for $\boldsymbol{S}^{(x)}$. Typically $\frac{1}{2}m(m+1) > n$ and equation (5.3) is thus overdetermined. The least square estimate (LSE) solution (see e.g. [32]) to the equation is

$$\boldsymbol{S}^{(x)} = (\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}. \tag{5.4}$$

## 5.3 Projection Method

Now that we have an estimate for the variances of each OD-pair, it is trivial to find an estimate of the mean by using the mean-variance relation (5.2).

$$\boldsymbol{\lambda}_0 = (\phi^{-1}\boldsymbol{S}^{(x)})^{\frac{1}{c}}. \tag{5.5}$$

The problem with this estimate is that it does not require the solution to satisfy the link count equation (1.4), which is a stronger condition than the second moment relation. The preliminary estimate $\boldsymbol{\lambda}_0$ can be improved by projecting the result to the surface that satisfies the link count condition.[1] This yields our estimate

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}(\overline{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{\lambda}_0). \tag{5.6}$$

Compared to the maximum likelihood approach, we do the moment estimation sequentially: First obtaining an estimate for the covariance matrix and then solving for

[1] Although the traffic matrix $\boldsymbol{A}$ is not necessarily of sufficient rank so that the inversion in the below formula can be done, we can always prune the access links, so that the rank of the matrix becomes the same as the number of links.

the mean. This does not yield quite as accurate estimates as MLE, but calculating it is many times faster.

The projection method works for any fixed parameters $\phi$ and $c$. In fact, we can try to estimate these parameters by requiring that $\boldsymbol{\lambda}_0$ comes as close as possible to satisfying (1.4), i.e. that they minimize

$$\begin{aligned} f(\phi, c) &= (\overline{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{\lambda}_0)^{\mathrm{T}}(\overline{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{\lambda}_0) \\ &= (\overline{\boldsymbol{y}} - \boldsymbol{A}(\phi^{-1}\boldsymbol{S})^{\frac{1}{c}})^{\mathrm{T}}(\overline{\boldsymbol{y}} - \boldsymbol{A}(\phi^{-1}\boldsymbol{S})^{\frac{1}{c}}). \end{aligned} \tag{5.7}$$

The values of $\phi$ and $c$ that realize the minimum, can now be used in equation (5.5) to yield the estimate (5.6) for $\boldsymbol{\lambda}$.

### 5.3.1 Estimating Parameters $\phi$ and $c$

In Cao et al. [5] the EM algorithm is run after preselecting a value for the exponent $c$ in the power law relation (5.2), while $\phi$ remains a parameter that the algorithm optimizes. The authors point out that convergence to a positive solution is guaranteed for the algorithm for integer values of $c$, namely $1$ or $2$.

Gunnar et al. [12] in their study of the Global Crossing data find out that the values for $c$ in those particular networks are $1.5$ and $1.6$ for the European and North American core-networks, respectively. Thus being limited to integer values in the solution makes sense for only computational reasons. The projection method works for any preselected $c$. In fact, we can relax $c$ to be a free parameter, though this means that we will no longer be able to obtain a closed form solution.

Minimization of (5.7) with respect to $\phi$ is a simple quadratic problem. So we can easily find the minimizing value $\widehat{\phi}(c)$. Now we can either use a preselected value for $c$ to yield the optimal $\phi$ value, or insert $\widehat{\phi}(c)$ back to (5.7), which yields

$$f(\widehat{\phi}(c), c) = (\overline{\boldsymbol{y}} - \boldsymbol{A}(\widehat{\phi}(c)^{-1}\boldsymbol{S})^{\frac{1}{c}})^{\mathrm{T}}(\overline{\boldsymbol{y}} - \boldsymbol{A}(\widehat{\phi}(c)^{-1}\boldsymbol{S})^{\frac{1}{c}}). \tag{5.8}$$

Now we have a simple one-parameter numerical minimization to find the optimal value of $c$. Expression (5.8) as a function of $c$ is depicted in Figure 5.1. The figure was generated by a set of synthetic data using parameter value $c = 1.5$.

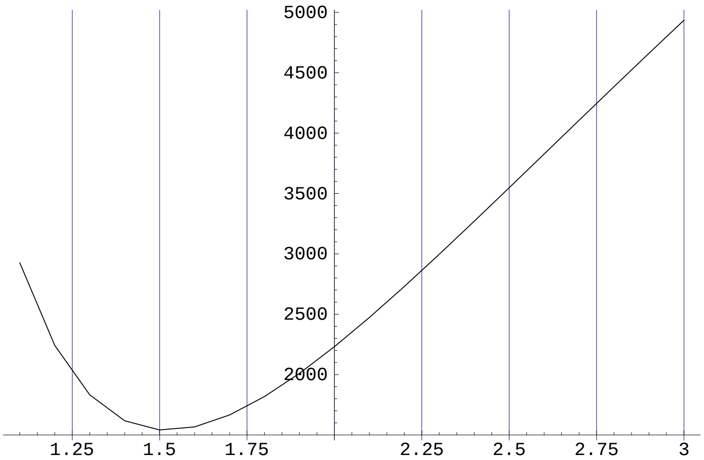


Figure 5.1: Values of the objective function (5.8) as a function of parameter $c$.

## 5.4 Constrained Minimization

Another approach is to require the condition $\overline{\boldsymbol{y}} = \boldsymbol{A\lambda}$ to be satisfied from the outset, and try to satisfy the mean-variance relation in the least square sense. In general, this has to be solved numerically. However, in the special case of $c = 1$ an explicit solution can be derived.

This approach is equivalent to Vardi's method, if we set $\epsilon$ very small, so that the first moment is the dominant factor in the estimation, with the exception that we treat $\phi$ as a parameter to be optimized, whereas in (3.23) it is fixed to $1$ by the Poisson assumption.

We get a constrained minimization problem

$$\begin{aligned} \min_{\lambda,\phi} \quad & \|\boldsymbol{S}^{(y)} - \boldsymbol{B}\phi\boldsymbol{\lambda}^c\| \\ \text{subject to} \quad & \overline{\boldsymbol{y}} = \boldsymbol{A\lambda}. \end{aligned} \tag{5.9}$$

Introducing a vector of Lagrange multipliers $\boldsymbol{\alpha}$, the objective function to be minimized can be written as

$$\begin{aligned} f(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \phi) &= (\boldsymbol{S}^{(y)} - \phi\boldsymbol{B\lambda})^{\mathrm{T}}(\boldsymbol{S}^{(y)} - \phi\boldsymbol{B\lambda}) + 2\boldsymbol{\alpha}^{\mathrm{T}}(\overline{\boldsymbol{y}} - \boldsymbol{A\lambda}) \\ &= \phi^2\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B\lambda} - 2\phi{\boldsymbol{S}^{(y)}}^{\mathrm{T}}\boldsymbol{B\lambda} - 2\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{A\lambda} + {\boldsymbol{S}^{(y)}}^{\mathrm{T}}\boldsymbol{S}^{(y)} + 2\boldsymbol{\alpha}^{\mathrm{T}}\overline{\boldsymbol{y}}. \end{aligned} \tag{5.10}$$

The above expression is quadratic in $\boldsymbol{\lambda}$, and the minimum with respect to $\boldsymbol{\lambda}$ can easily be found,

$$\boldsymbol{\lambda} = \phi^{-2}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\alpha} + \phi\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}). \tag{5.11}$$

The Lagrange multipliers $\boldsymbol{\alpha}$ are then determined such that the constraints are satisfied:

$$\overline{\boldsymbol{y}} = \boldsymbol{A}\phi^{-2}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\alpha} + \phi\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}), \tag{5.12}$$

from which

$$\boldsymbol{\alpha} = (\phi^{-2}\boldsymbol{A}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{A}^{\mathrm{T}})^{-1}(\overline{\boldsymbol{y}} - \phi^{-1}\boldsymbol{A}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}). \tag{5.13}$$

Minimizing $f(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \phi)$ with respect to $\phi$ yields

$$\phi = (\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\lambda})^{-1}{\boldsymbol{S}^{(y)}}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\lambda}. \tag{5.14}$$

Substitution of (5.13) into (5.11) gives $\boldsymbol{\lambda}$ as a function of $\phi$

$$\boldsymbol{\lambda} = \boldsymbol{K}\overline{\boldsymbol{y}} - \phi^{-1}(\boldsymbol{K}\boldsymbol{A}(\boldsymbol{B}^{T}\boldsymbol{B})^{-1}\boldsymbol{B}^{T}\boldsymbol{S}^{(y)} + \boldsymbol{B}^{T}\boldsymbol{S}^{(y)}),$$

where we use the notation

$$\boldsymbol{K} = (\boldsymbol{B}^{T}\boldsymbol{B})^{-1}\boldsymbol{A}^{T}(\boldsymbol{A}(\boldsymbol{B}^{T}\boldsymbol{B})^{-1}\boldsymbol{A}^{T})^{-1}.$$

Substituting $\boldsymbol{\lambda}$ further in (5.14) yields an quadratic equation for $\phi$, which is easily solvable. This solution can be then substituted back to (5.13) and (5.11) to obtain the explicit expression for $\boldsymbol{\lambda}$.

## 5.5 Comparison with the MLE Method

The accuracy of the quick methods are evaluated by comparing them against Maximum likelihood estimation, presented in section 3.4. In the subsequent sections the results of accuracy on synthetic data test cases is presented.

### 5.5.1 Results

For the evaluation we use two topologies. A small six node topology, shown in Figure 5.2, has $14$ one-way links, two links between each connected pair of node. Assuming traffic from each node to all other nodes, there are $30$ OD pairs in the network. In the more realistic size fictitious backbone topology shown in Figure 5.4, there are $12$ nodes, $38$ links, and $132$ OD pairs. For both topologies, we generate synthetic Gaussian data sets, where the power-law holds. Sample size is set to $500$ measurements for each simulation.

### A Simple Six Node Topology

In the synthetic OD pair traffic that we use, the traffic varies so that the largest OD pairs are ten times as large as the smallest ones.

Figure 5.3 illustrates the results for the maximum likelihood estimates, projection method, the constrained minimization, and Vardi's method solved with the least square method, which we call here "Quick Vardi". The synthetic data used for the evaluations is generated with parameters $c = 1$, $\phi = 1$. This is equivalent to the Poisson-assumption made in Vardi's method.

The OD pairs are presented in ascending order based on the traffic amount, so that the smaller OD pairs are on the left and the largest on the right. We see that, as expected, the MLE performs better on average, but not overwhelmingly better. The average errors are $15\%$, $26\%$, $34\%$ and $35\%$ for the MLE, the projection method, constrained minimization and Quick Vardi respectively.

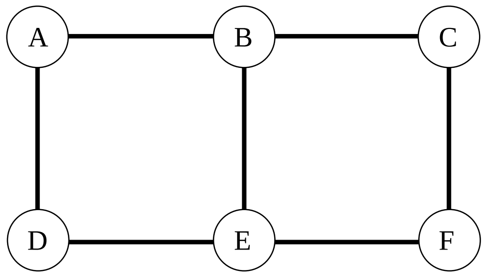


Figure 5.2: Six node Test topology

### A 12 Node Backbone Topology

In this example we use synthetic data generated with parameter value $c = 1.5$. The traffic volumes for the OD pairs vary so that the largest are approximately hundred times as large as the smallest ones. This creates great difficulties for the quick methods regarding the estimation of the smaller OD pairs. The estimates of the projection method for the smallest OD pairs are far off the real traffic amounts. Due to the fact that the estimates for some of the smallest OD pairs have errors of several hundred percent, the mean error is also affected greatly by these, and is $59\%$ for the projection method and $110\%$ for the constrained optimization, while it is $29\%$ for the MLE. The mean error for the Quick Vardi method is several hundred percent, so it is not considered here.

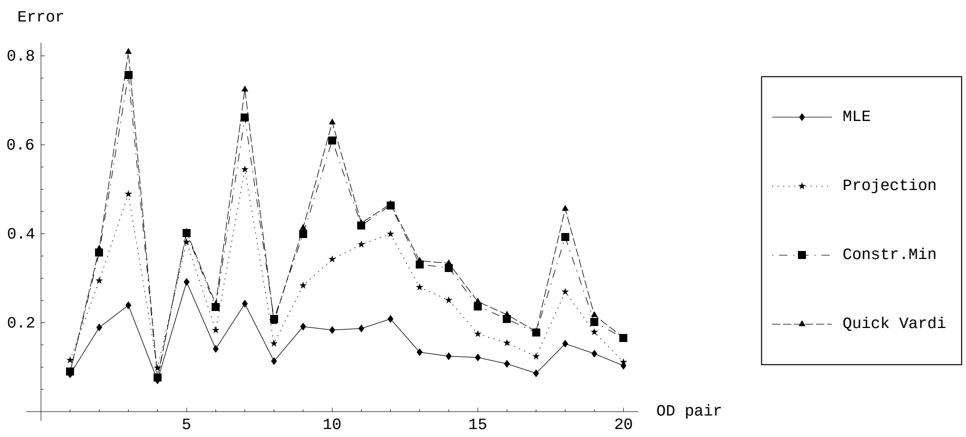


Figure 5.3: Errors for OD pairs in $6$-node topology in ascending order of traffic amount for case $c = 1$.

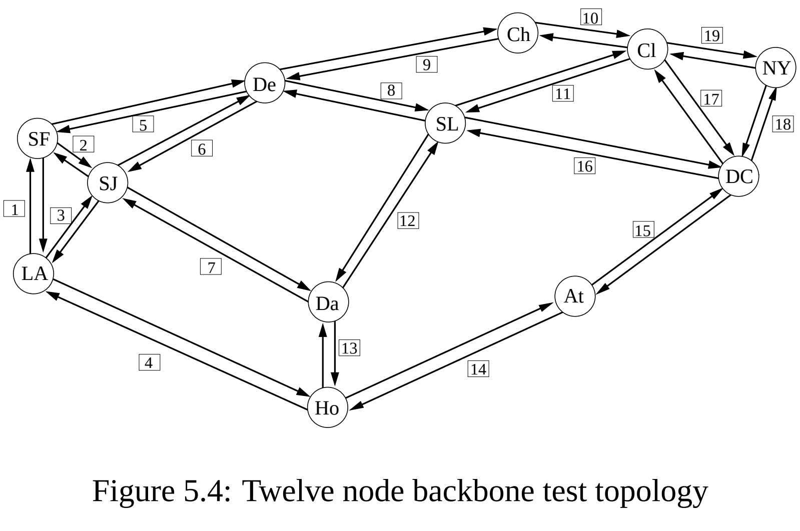


Figure 5.4: Twelve node backbone test topology

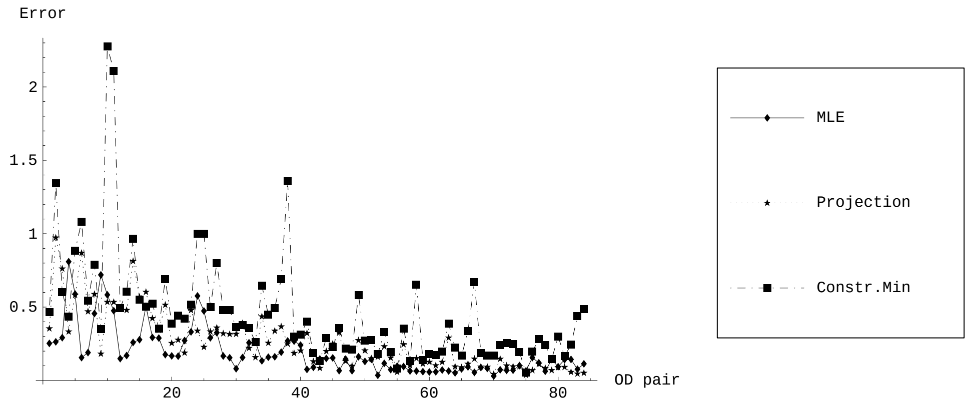


Figure 5.5: Errors for largest OD pairs in 12-node topology in ascending order of traffic amount

However, the most important thing is to estimate the largest OD pairs. If we concentrate only on the largest OD pairs that comprise $90\%$ of total traffic in volume, the projection method is more competitive. The errors for these OD pairs are shown in Figure 5.5. The mean errors are $27\%$ for the projection method, $46\%$ for the constrained minimization and $19\%$ for the MLE.

# Chapter 6

# Conclusion

*In this chapter a summary of the thesis is given and directions for further work are discussed. The chapter concentrates on the findings of the literature review and on the Quick method proposed in chapter 5.*

## 6.1 Summary

While the traffic matrix is a crucial input in many network planning and traffic engineering tasks, it is usually not possible to directly measure it. Although measuring tools such as Netflow are becoming more common, the measurements still increase cost and overhead in the network and are thus not necessarily readily obtainable. In the classical traffic matrix estimation framework, on which this thesis concentrated, the only available information is considered to be the link count measurements and the routing table.

As in any realistic network there are more OD pairs than links, the problem is highly underconstrained and thus ill-posed. This means that exact solutions cannot be obtained. To get an estimate, some extra information has to be brought in to the situation. Based on a comprehensive literature review we found that while there are altogether almost twenty different methods proposed for the traffic matrix estimation problem, an overwhelming majority of these fall into two main categories with regard to the nature of the extra information:

1. Gravity model based methods.

2. Methods using second moment statistics through a mean-variance relation.

The first group of estimation methods uses the gravity model assumption to gain the extra information. The gravity model assumes that the traffic between source node $s$ and destination node $d$ is directly proportional to the product of total traffic sent by $s$ and total traffic received by $d$. Based on this model it is possible to form a prior estimate. This information is then combined with the link count information to yield the final estimate.

The second group brings the extra information from the link count sample covariance matrix. When a functional relation is assumed between the mean and the variance of OD counts, it is possible to formulate a maximum likelihood problem that becomes identifiable through the use of the sample covariance.

The accuracy of these methods depend on the validity of the assumptions. The first group of methods make only the gravity model assumptions. This is found to be accurate for some networks but inaccurate for others, see, e.g., [12]. The second group makes the assumption of a functional relation existing between mean and variance. In addition, a traffic distribution has to be assumed to formulate the maximum likelihood equation. Poisson distribution has been proposed, but the results were not encouraging, and commonly the Gaussian distribution is used for this purpose. We studied these two assumptions in chapter 2 and found them adequate, yet not perfect, fits with regard to the Funet dataset used.

The lack of an extensive evaluation of estimation approaches in current literature makes it difficult to assess the accuracy of the methods. As far as we are aware of, there is no evaluation with real data between methods from the two different groups of estimation methods.

A study with synthetic data concentrating on the sensitivity analysis of the assumptions specified above should prove beneficial for the evaluation of the methods and overall understanding of the problem. We did a small simulation study in this area in section 2.4.2 to explore the effects of inaccuracy in the mean-variance relation. A more detailed analysis including the gravity model assumption and Gaussian distribution assumption is left as future work.

## 6.2 On the Quick Method

In chapter 5 we proposed two techniques to obtain an estimate for traffic matrix by explicit calculations utilizing the link count covariance matrix. We illustrated how one can obtain the OD pair traffic variance estimates from empirical link count covariance matrix, and developed computationally light weight methods, the projection method and the constrained minimization method, to obtain an estimate for the traffic matrix based on the link count covariance matrix, in a way that would still be consistent with the link counts.

The constrained minimization method was recognized, in fact, to be a special case of Vardi's method. We gave an explicit solution for it in the case $c = 1$ and also obtained an estimate for the second parameter $\phi$ in the mean-variance relation. For the projection method we have an even simpler and quicker way to compute solution. Also in this case we get estimates of the parameters $c$ and $\phi$.

We evaluated the accuracy of the methods in a simulation study by comparing them against the maximum likelihood solution by Cao et al. [5] and found that they perform reasonably well, considering that they are much quicker and simpler to calculate than the MLE, which requires the use of an iterative numerical method, namely the EM algorithm. In the worst case, the errors in the estimate of a traffic matrix element for the largest components given by the quick method were three times as large as those by the MLE method, but in many cases the difference was smaller. As for the running time, the difference between the MLE method and quick methods was big. With our non-optimized Mathematica code running the MLE method took of the order of minutes, while the quick methods yielded the result in less than a second.

All the comparisons were done with synthetic data. Evaluation with real data would be very important to assess the true effectiveness of the methods. For now, we have used in our evaluations a sample size of $500$, which may be rather large in comparison to what is available in reality. Accuracy of the estimated covariance matrix with various sample sizes should be studied, as well as the effect the measurement inaccuracies have on the subsequent traffic matrix estimates.

# Bibliography


[1] I. Juva, R. Susitaival, M. Peuhkuri, and S. Aalto, "Traffic characterization for traffic engineering purposes: Analysis of Funet data", NGI 2005, Rome, Italy, April 2005.

[2] R. Susitaival, I. Juva, M. Peuhkuri, and S. Aalto, "Characteristics of OD pair traffic in Funet" ICN'06, Mauritius, April 2006.

[3] I. Juva, S. Vaton, J. Virtamo, "Quick Traffic Matrix Estimation Based on Link Count Covariances", in EURO NGI Workshop on Traffic Engineering, Protection and Restoration, Rome, Italy, 2005.

[4] I. Juva, P. Kuusela, J. Virtamo, "A Case Study on Traffic Matrix Estimation Under Gaussian Distribution", in Seventeenth Nordic Teletraffic Seminar, Fornebu, Norway, 2004.

[5] J. Cao, D. Davis, S. V. Wiel, B. Yu, "Time-varying network tomography," Journal of the American Statistical Association, vol. 95, 2000.

[6] J. Cao, S. V. Wiel, B. Yu, Z. Zhu, "A scalable method for estimating network traffic matrices," Bell Labs, Technical Report, 2001.

[7] P. Carlsson, M. Fiedler, K. Tutschku, S. Chevul and A. Nilsson, "Obtaining Reliable Bit Rate Measurements in SNMP-Managed Networks", COST 279TD(02)21.

[8] J.D. Case, M. Fedor, M.L. Schoffstall and C. Davin, "Simple Network Management Protocol (SNMP)", May 1990, RFC1157, STD0015.

[9] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet tomography", IEEE Signal Processing Magazine, May 2002.

[10] S. Eum, J. Murphy, R.J. Harris, "A Fast Accurate LP Approach for Traffic Matrix Estimation", in ITC19, Beijing, China, 2005.

[11] O. Goldschmidt, "ISP Backbone Traffic Inference Methods to Support Traffic Engineering", In Internet Statistics and Metrics Analysis Workshop, San Diego, 2000.

[12] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large IP backbone – A comparison on real data", in IMC'04, Taormina, Italy, October 2004.

[13] J. Kowalski and B. Warfield, "Modeling traffic demand between nodes in a telecommunications network", ATNAC 95, 1995.

[14] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E.D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows", in SIGMETRICS/Performance'04, New York, USA, June 2004.

[15] G. Liang, B. Yu, "Pseudo Likelihood Estimation in Network Tomography", in Infocom 2003.

[16] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new solutions", in SIGCOMM'02, Pittsburg, USA, August 2002.

[17] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, and C. Diot, "A Taxonomy of IP Traffic Matrices", in SPIE ITCOM: Scalability and Traffic Control in IP Networks II, Boston, August 2002.

[18] A. Medina, K. Salamatian, N. Taft, I. Matta, Y. Tsang, C. Diot, "On the Convergence of Statistical Techniques for Inferring Network Traffic Demands", Tecnhical Report, 2003.

[19] A. Medina, K. Salamatian, N. Taft, I. Matta, C. Diot, "A Two-step Statistical Approach for Inferring Network Traffic Demands", Tecnhical Report, 2004.

[20] A. Nucci, R. Cruz, N. Taft, C. Diot, "Design of IGP Link Weight Changes for Estimation of Traffic Matrices", IEEE Infocom, Hong Kong, China, March 2004.

[21] K. Papagiannaki, N. Taft, A. Lakhina, "A Distributed Approach to Measure Traffic Matrices", in ACM Internet Measurement Conference, Taormina, Italy, 2004.

[22] M. Peuhkuri, "A method to compress and anonymize packet traces", in IMW'01, San Francisco, USA, 2001.

[23] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft, How to identify and estimate the largest traffic matrix elements in a dynamic environment, in SIGMETRICS/Performance'04, New York, USA, June 2004.

[24] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, Traffic matrices: Balancing measurements, inference and modeling, in SIGMETRICS'05, Banff, Canada, June 2005.

[25] A. Soule, K. Salamatian, A. Nucci, N. Taft, "Traffic Matrix tracking using Kalman Filtering", Technical Report, 2004.

[26] C. Tebaldi and M. West, "Bayesian inference on network traffic using link count data", Journal of American Statistical Association, 93, pp. 557–576, 1998.

[27] Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data", Journal of the American Statistical Association, vol. 91, pp. 365–377, 1996.

[28] S. Vaton, A. Gravey, "Network tomography: an iterative Bayesian analysis", ITC18, Berlin, August 2003.

[29] S. Vaton, J.S. Bedo, A. Gravey, "Advanced methods for the estimation of the Origin Destination traffic matrix", Revue du 25ème anniversaire du GERAD, 2005.

[30] Y. Zhang, M. Roughan, N. Duffield, A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads", ACM Sigmetrics 2003.

[31] Y. Zhang, M. Roughan, C. Lund, D. Donoho, "An Information-Theoretic Approach to Traffic Matrix Estimation" ACM SIGCOMM 2003.

[32] David G. Luenberger, "Optimization by Vector Space Methods", New York NY, John Wiley & Sons, 1969.

[33] G. J. McLachlan and T. Krishnan, "The EM Algorithm and Extensions", John Wiley and Sons, 1997.

[34] J. S. Milton, J. C. Arnold, "Introduction to Probability and Statistics", McGraw-Hilh Inc. , 1995.

[35] C. Robert, G. Casella, "Monte Carlo Statistical Methods", Springer, 1999.

[36] Cisco NetFlow, www.cisco.com/warp/public/732/Tech/netflow.

# Appendix A

# Deriving the EM Equations

To derive equations (3.32) and (3.33) we need to use the following

**Matrix inversion lemma**

$$(\boldsymbol{A}+\boldsymbol{BCD})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{DA}^{-1}\boldsymbol{B}+\boldsymbol{C}^{-1})^{-1}\boldsymbol{DA}^{-1}, \tag{A.1}$$

$$\boldsymbol{y} = \boldsymbol{Ax} + \epsilon \qquad \epsilon \sim N(0, \sigma^2\boldsymbol{I}), \qquad \boldsymbol{x} \sim N(\boldsymbol{\lambda}, \boldsymbol{\Sigma}).$$

$$\begin{aligned}
p(\boldsymbol{x}) &\propto \exp[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\lambda})]], \\
p(\boldsymbol{y}|\boldsymbol{x}) &\propto \exp[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{Ax})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{Ax})], \\
p(\boldsymbol{x}|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}), \\
\log p(\boldsymbol{x}|\boldsymbol{y}) &= \text{constant} - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\lambda}) - \frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{Ax})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{Ax}) \\
&= \text{cnst.} - \frac{1}{2}[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - 2\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \frac{1}{\sigma^2}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax} - 2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{Ax})] \\
&= \text{cnst.} - \frac{1}{2}[\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{x} - 2(\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{x}] \qquad \text{(A.2)} \\
\log p(\boldsymbol{x}|\boldsymbol{y}) &= \text{cnst.} - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}(\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})(\boldsymbol{x}-\boldsymbol{\mu}). \qquad \text{(A.3)}
\end{aligned}$$

## Deriving $\mathrm{Var}[\boldsymbol{x}|\boldsymbol{y}]$

So now from (A.2) and (A.3) we can pick out the variance as the term that depends on $\boldsymbol{x}^T\boldsymbol{x}$

$$\mathrm{Var}[\boldsymbol{x}|\boldsymbol{y}] = [\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}]^{-1}.$$

Then using the Matrix inversion lemma, with

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{\Sigma}^{-1}, \\ \boldsymbol{B} &= \boldsymbol{A}^{\mathrm{T}}, \\ \boldsymbol{C} &= \frac{1}{\sigma^2}, \boldsymbol{I} \\ \boldsymbol{D} &= \boldsymbol{A}. \end{aligned}$$

yields

$$\begin{aligned} \mathrm{Var}[\boldsymbol{x}|\boldsymbol{y}] &= \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\ \boldsymbol{R} = \lim_{\sigma\to 0}\mathrm{Var}[\boldsymbol{x}|\boldsymbol{y}] &= \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}. \end{aligned} \tag{A.4}$$

## Deriving E[x|y]

As in previous section we use (A.2) and (A.3) to write

$$\boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})x = (\boldsymbol{\lambda}^{\mathrm{T}}\Sigma^{-1} + \frac{1}{\boldsymbol{\sigma}^2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{x}, \tag{A.5}$$

$$\begin{aligned} \boldsymbol{\mu}^{\mathrm{T}} &= (\boldsymbol{\lambda}^{\mathrm{T}}\Sigma^{-1} + \frac{1}{\boldsymbol{\sigma}^2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A})(\boldsymbol{\Sigma}^{-1} + \frac{1}{\sigma^2}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1} \\ &= (\boldsymbol{\lambda}^{\mathrm{T}}\Sigma^{-1} + \frac{1}{\boldsymbol{\sigma}^2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A})(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\ &= (\boldsymbol{\lambda}^{\mathrm{T}} + \frac{1}{\boldsymbol{\sigma}^2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{\Sigma})(\boldsymbol{I} - \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\ &= \boldsymbol{\lambda}^{\mathrm{T}} - (\boldsymbol{A}\boldsymbol{\lambda})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\ &\quad + \frac{1}{\sigma^2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{\Sigma}(\boldsymbol{I} - \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}. \end{aligned} \tag{A.6}$$

Now using the matrix inversion lemma with

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}, \\ \boldsymbol{B} &= \sigma^2\boldsymbol{I}, \\ \boldsymbol{C} = \boldsymbol{D} &= \boldsymbol{I}, \end{aligned}$$

we can write

$$(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1} \quad = \quad (\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{I} - \sigma^2(\sigma^2(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1} + \boldsymbol{I})^{-1}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}).$$

Then concentrating on the second term of the sum in (A.6) and using the above result

$$\begin{aligned}
&\frac{1}{\sigma^2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{\Sigma}(\boldsymbol{I} - \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} = \\
&\qquad = \frac{1}{\sigma^2}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma} - \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}) \\
&\qquad = \frac{1}{\sigma^2}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma} - (\boldsymbol{I} - \sigma^2(\sigma^2(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1} + \boldsymbol{I})^{-1}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1})\boldsymbol{A}\boldsymbol{\Sigma}) \\
&\qquad = \frac{1}{\sigma^2}\boldsymbol{y}^{\mathrm{T}}(\sigma^2(\boldsymbol{I} + \sigma^2(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1})^{-1}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}) \\
&\qquad = \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{I} + \sigma^2(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1})^{-1}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}.
\end{aligned}$$

And then continuing from (A.6) while inserting this to the second term of the sum yields

$$\begin{aligned}
\boldsymbol{\mu}^{\mathrm{T}} &= \boldsymbol{\lambda}^{\mathrm{T}} - (\boldsymbol{A}\boldsymbol{\lambda})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\
&\quad + \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{I} + \sigma^2(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1})^{-1}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\
\lim_{\sigma\to 0}\boldsymbol{\mu}^{\mathrm{T}} &= \boldsymbol{\lambda}^{\mathrm{T}} - (\boldsymbol{A}\boldsymbol{\lambda})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} + \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\
&= \boldsymbol{\lambda}^{\mathrm{T}} + (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\lambda})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma} \\
\boldsymbol{m} = \lim_{\sigma\to 0}\boldsymbol{\mu} &= \boldsymbol{\lambda} + \boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\lambda}). \qquad \text{(A.7)}
\end{aligned}$$