How I Learned to Stop Worrying and Love Traffic Matrices

Prof. Matthew Roughan
matthew.roughan@adelaide.edu.au
http://www.maths.adelaide.edu.au/matthew.roughan/

UoA

April 4, 2016

Acks

Many people have helped with the work described here, here's just a few in no particular order

- Albert Greenberg
- Yin Zhang
- Paul Tune
- Walter Willinger
- Zihui Ge
- Jen Yates
- Nick Duffield
- Anja Feldman
- Carsten Lund
- David Donoho
- Mikkel Thorup

and many others.

Additional Information

- More complete notes can be found in **Internet Traffic Matrices: A Primer**, Paul Tune and Matthew Roughan, ACM Sigcomm eBook, “Recent Advances in Networking SIGCOMM eBook”, Vol.1, August 2013.
  www.sigcomm.org/content/ebook
- My web page
  www.maths.adelaide.edu.au/matthew.roughan
  - Slides available at
    /talks.html
  - Links, and some data and code available
    /traffic_matrices.html

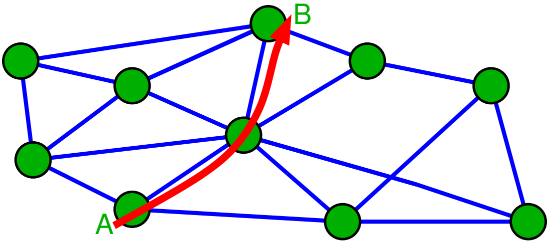
The Hard Lesson (for me)

- This is a little story about a Boy from the Sticks who went to the Big City and ...
- Queueing Theory wasn't as useful as I thought
  - My PhD was on Queueing Theory (quite a while ago)
  - I went to work at AT&T Labs (which to me was the Mecca of queueing theory) and no-one cared
- Why?
  - unrealistic assumptions
  - not solving a real problem
  - **lack of data**
    - and the data they did have wasn't what you needed
- So it turns out the maths I knew was solving the wrong problem
  - the real problems were actually easier!
  - but less specific, so I had to learn a wider skill base
- A big part revolved around **traffic matrices**, so here, now live, for one night only ...

Section 1

Intro

Traffic Matrix [1]



$$T = \begin{pmatrix} t_{AA} & t_{AB} & t_{AC} & \cdots \\ t_{BA} & t_{BB} & t_{BC} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

t_{AB} is the traffic going from location A to location B across a network

Taxonomy of Traffic Matrices

- In today's network locations can be
  - physical
    - PoPs = Points of Presence
    - routers
    - links
    - servers
  - logical
    - IP addresses
    - common-prefix address blocks (prefixes)
- Offered vs Carried Load
  - Demand vs Traffic matrix
- Ingress-Egress (IE) vs Origin-Destination (OD)

Offered vs Carried Load

- Offered load is the potential traffic
  - traffic desire network
- Carried load is what we actually see
- They can be different
  - congestion (*i.e.,* capacity constraints reached)
  - feedback (formal or heuristic)
  - non-locality: we observe at a point distant from origin and destination
  - anomalies (*e.g.,* a car crash, link outage, ...)
- It's quite hard to observe offered load so mostly we talk about traffic matrices, not demand matrices

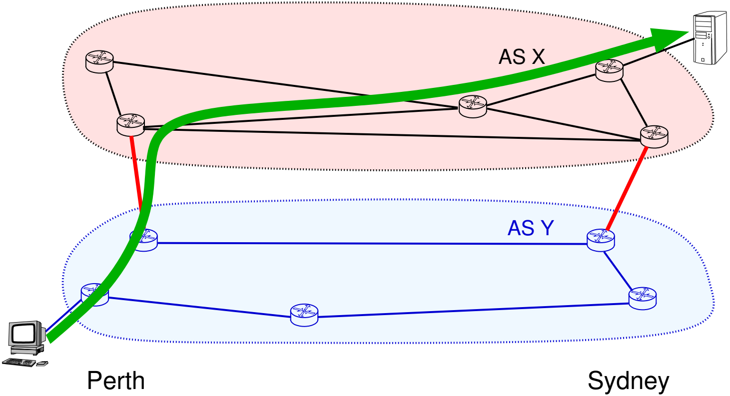
Invariants

Why do these issues really matter?

- What we really want is an **invariant**
  - under some set of changes to the network, it doesn't change
  - *e.g.,* in optimisation, we want an input that is invariant under changes in the optimisation variables, within the constraints
- We rarely have a true invariant
  - offered load is more useful than carried load
  - even offered loads aren't completely invariant
    - *e.g.,* new roads change housing patterns, and hence traffic
    - *e.g.,* **IE traffic matrices**

Hot Potato Routing

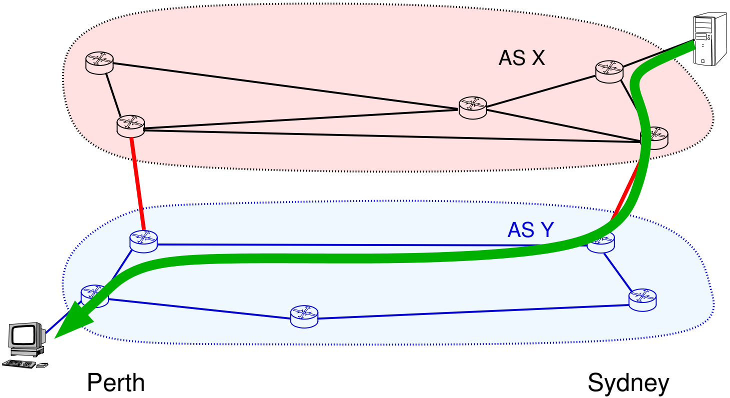
- dump traffic off your network as fast as possible



- traffic from Perth on AS Y to Sydney on AS X

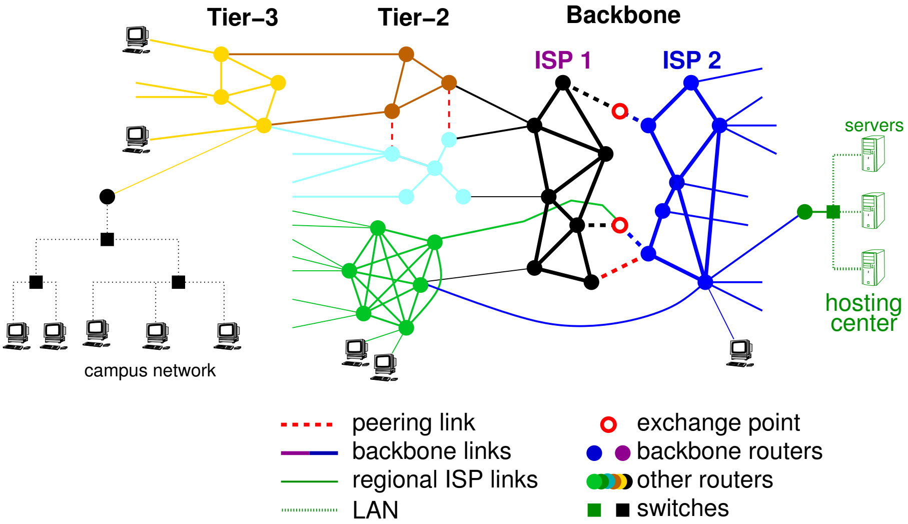
Hot Potato Routing

- dump traffic off your network as fast as possible



- If I run AS X, all I see is traffic on my network!
  - IE Traffic Matrices are asymmetric!
  - IE Traffic Matrices are subject to the whims of routing

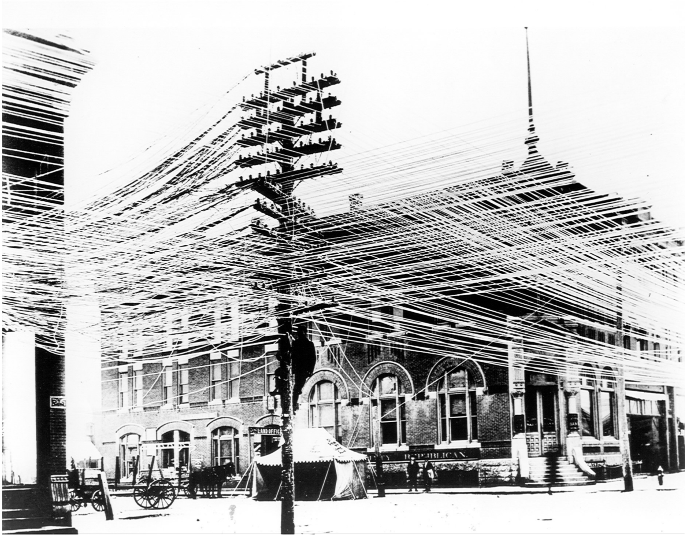
A Cartoon of the Internet



Why do we care about TMs and invariants?

Why do we care about TMs and invariants?

Dumb network design (Pratt, Kansas, c1900)



http://www.bellsystemmemorial.com/oldphotos_6.html

TM Applications

- Network Operators
  - Network Planning (optimisation)
    - capacity planning (green-fields or incremental)
    - traffic engineering
    - ...
  - Network Reliability Analysis
  - Anomaly Detection

  These need a **predicted** **real** traffic matrix
- Researchers
  - Protocol Design
    - *e.g.,* routing protocols
  - Algorithm Design
    - *e.g.,* traffic engineering optimisation algorithms

  These need an **ensemble** of **controllable** TMs

TM Time-line

1937	Telephone Traffic	Kruithof [2, 3]
1960s	Transportation Traffic	*e.g.*, [4]
1996-2000	Network Tomography	Vardi [5], followed by [6, 7]
2000+	Internet Measurement	Feldmann *et al.* [8, 9]
2002-10	Internet Tomography	Sprint v AT&T
		[10, 11, 12, 13, 14, 15, 16, 17, 18,
2004-10	Anomaly Detection	[22, 23, 24, 25, 20]
2005+	Synthesis	[26, 27]
⋮		

we'll add some more recent bits towards the end of today's talk

Outline

Section 2

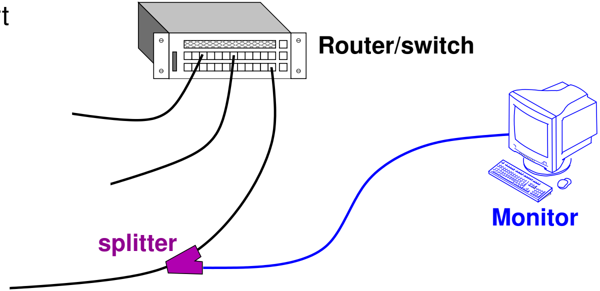
How do you get a TM?

How to Get Traffic Data

- Packet trace
- SNMP
- Inference (tomography)
- Netflow

Packet traces

- tap a link, or router
  - optical, or electronic splitter/coupler
  - monitoring port



- record every packet's
  - size
  - time (of first byte)
  - headers (IP, TCP, possibly more)

Data Volume Management

> Space is big. You just won't believe how vastly, hugely, mind-bogglingly big it is...
>
> *Douglas Adams,*
> *Hitchhiker's Guide to the Galaxy*

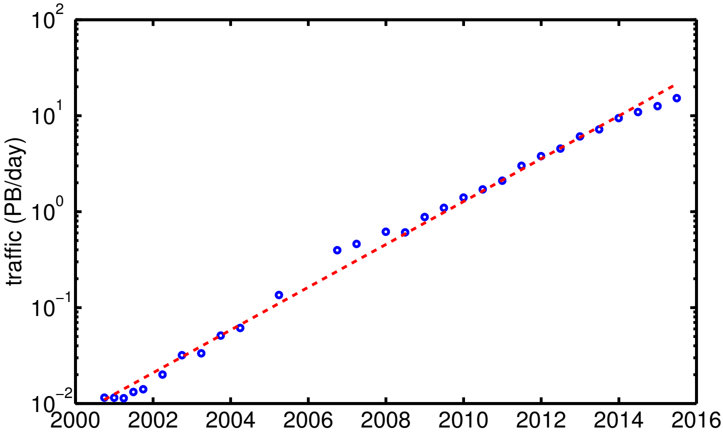
- Internet measurement was one of the first real sources of

  BIG DATA

  - well before the term became trendy
- 10 Gbps link generates 10 Gbps of traffic data (at peak)
  - 2 TB disk is full in less than half an hour
  - and a single 10 Gbps link isn't much today

Australian Traffic Growth

Doubling every $\sim$ 1.3 years



www.abs.gov.au

Data Volume Strategies

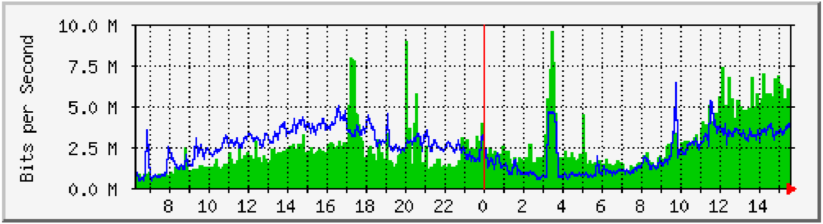
A number of operations can reduce the dataset size

- **sampling:**
  - standard statistical approach
  - simplest case, sample every Nth packet, or randomly choose 1 in N packets
- **filtering:** only look at packets which meet certain requirements, *e.g.*,
  - only TCP packets
  - only packets between two specific IP addresses
- **sketching**: (not today)
- **aggregation:** reduce the granularity of the data somehow
  - aggregate over time, or keys
  - examples: SNMP, Netflow

SNMP

Simple Network Management Protocol

- not just for management
- allows collection of traffic data



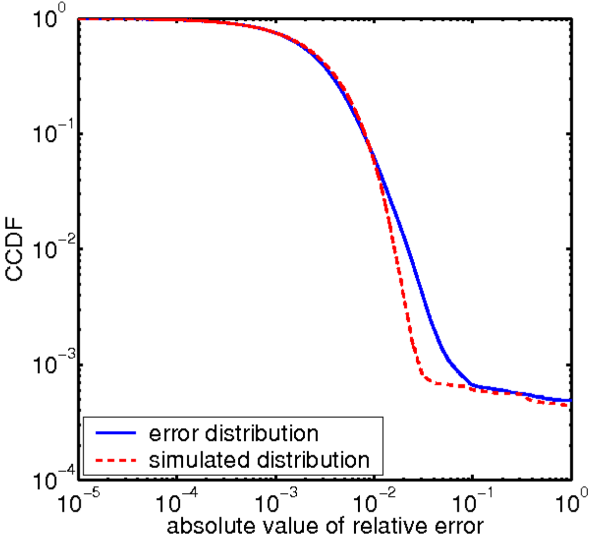
- but it's just crude counts
  - no details
  - coarse granularity (*e.g.*, 5 minutes)
  - error prone [28]
  - lots of missing data

Data Quality

If you think your data is clean, you haven't looked

- numerical errors
  - ALL data has numerical errors
    - our job is not completely remove errors
    - **calibrate**
    - no-one does that well (as yet)
- artifacts
  - the field isn't even a number, *e.g.,* NA
  - numbers in the wrong format, e.g, 1,000 in a CVS file
  - part of a file was overwritten by 2 processes writing to same file
  - a process crashed part way through writing the data
- missing data
  - large number of monitors, some will be offline all the time
- inconsistency
  - two DBs have different information
  - two DBs use different keys for same information
- ambiguous data
  - DB keys don't provide enough information for a task

Errors in SNMP [28]



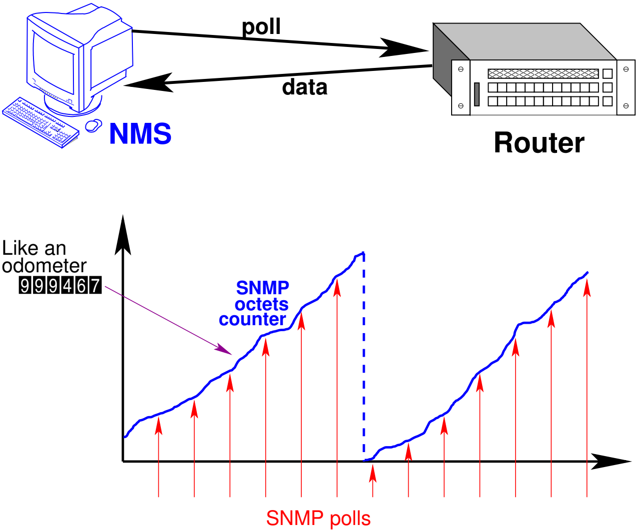
Errors in SNMP [28]

- Most errors are small
- Appears to be a heavy tail for larger errors
- Two main causes of errors
  - simulation is mixture of
    - exponential distribution with mean 0.0035 and probability 0.99882
    - Pareto distribution with cumulative distribution function

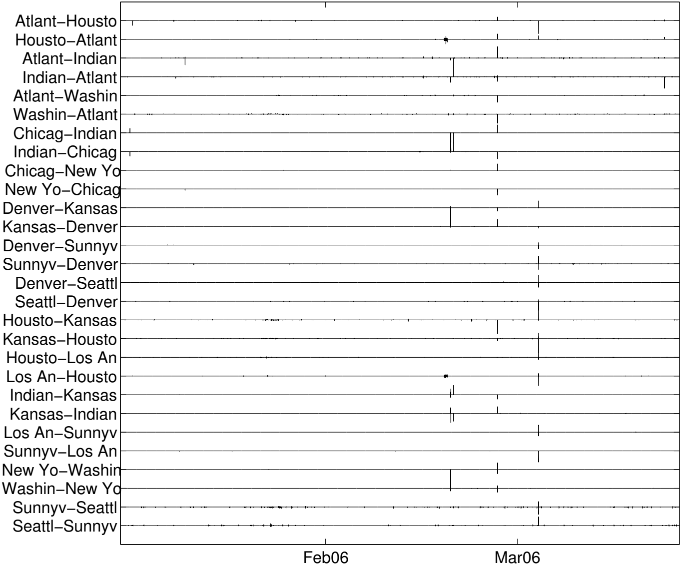
$$F(x) = 1 - \left(\frac{b}{x}\right)^{\alpha},$$

      with probability of selection of 0.00118, and parameters $\alpha = 0.12$ and $b = 0.0005$.
  - NB: Pareto component has infinite mean, so need large set of data to observe, and test it

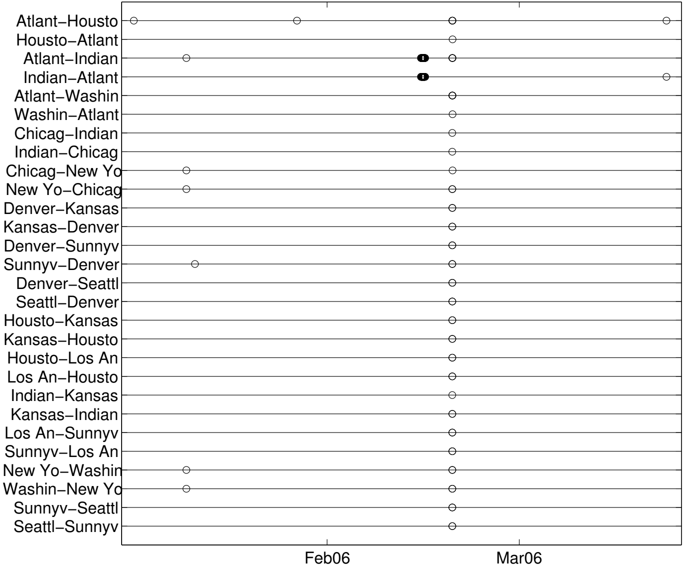
SNMP data collection



Errors correlations [28]



Missing Data in SNMP



SNMP and Traffic Matrices

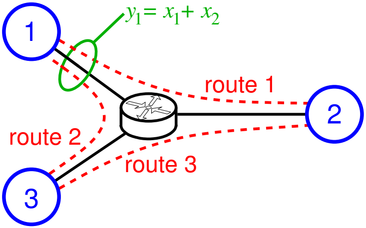
- SNMP contains link counts
  - packets per interface
  - bytes per interface
- No idea where the traffic is going!
  - it doesn't tell you the traffic matrix!

Network Tomography

Example

SNMP only gives link counts, not traffic matrices, but they are related

$$\boldsymbol{y} = R\boldsymbol{x}$$



$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = R\boldsymbol{x}$$

Network Tomography

Notes

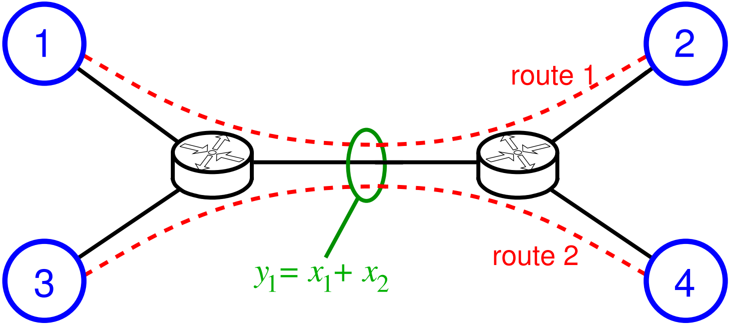
- Each of the columns of the matrix X are **stacked** to give a column vector $\boldsymbol{x}$
- Measurements have errors so

$$\boldsymbol{y} = R\boldsymbol{x} + \boldsymbol{z}$$

- R is not square, so we can't just invert it

Network Tomography

Another Example



$$\boldsymbol{y} = R\boldsymbol{x}$$

where $R = [1, 1]$

A Word on Routing Matrices

- What are they?
  - The matrix is an **incidence** matrix
  - The matrix is size $L \times N(N-1)$ where there are L links and N source/destinations in the network
  - Simplest form has 0 or 1s
  - A 1 in position (i, j) indicates that route j uses link i, where
  - Route i refers to a particular TM source/destination pair
  - With load balancing, the matrix might contain fractions
- How do I get one?
  - You need to know your network topology
    - ★ lots of ways to measure this
  - You need to know you network routing, either by
    - ★ measuring current forwarding paths
    - ★ measuring routing policies, and predicting routing

General Framework

Want to solve the **inverse** problem

$$\boldsymbol{y} = R\boldsymbol{x} + \boldsymbol{z}$$

but it's **highly-under-constrained**, so we need **side information** or a **model**, or a **prior**, then we solve via optimisation

$$\underset{\boldsymbol{x}}{\text{argmin}} \|\boldsymbol{y} - R\boldsymbol{x}\| + \lambda d(\boldsymbol{x}_m, \boldsymbol{x})$$

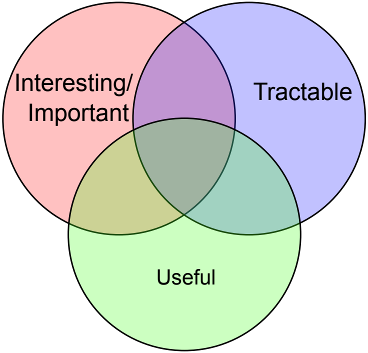
- Note we don't to force the equality because there are measurement errors
- General strategy is called regularisation
- Lots of different possible models
- λ lets you trade off between the distance $d(\cdot, \cdot)$ from the prior model $\boldsymbol{x}_m$ and the data $\boldsymbol{y}$
- You can use different norms $\|\cdot\|$ and distances

Network Tomography

Given stacked TM $\boldsymbol{x}$ and routing matrix R, the link loads on the network are given by $\boldsymbol{y}$ which can be written simply as

$$\boldsymbol{y} = R\boldsymbol{x} + \boldsymbol{z}$$

- lots research [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21],...
- why so much?



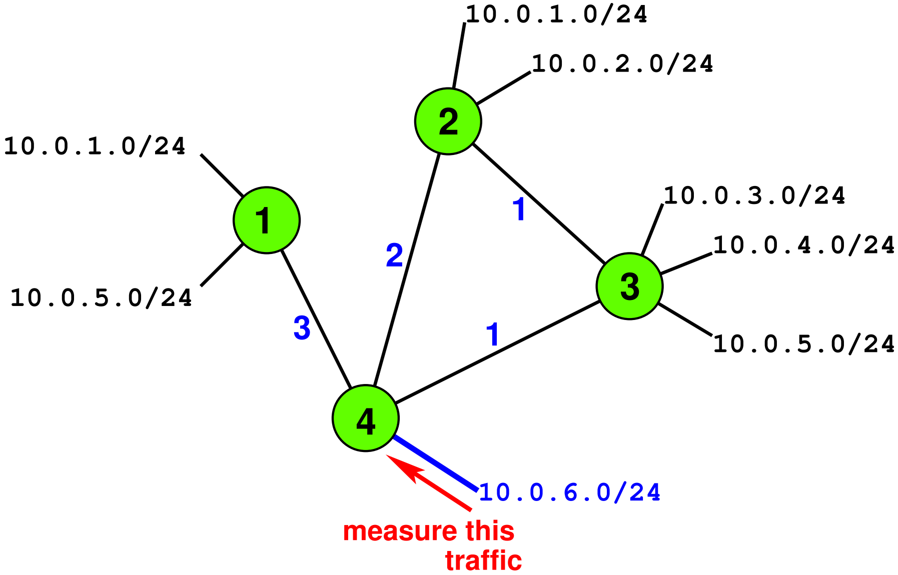
Why did TM Inference die out as a research topic?

- 2 sorts of data so far
  - packet trace – too hard to collect
  - SNMP – easy to collect, but hard to use
- There is a third sort: **Netflow**
  - it's been around for quite a while
  - but it wasn't very easy to collect until more recently

Netflow (Cisco v5)

- Idea: aggregate to close approximation of a TCP connection
  - keep one record per **flow**
  - **key** 5-tuple: IP source, dest, protocol and TCP source, dest port
  - also
    - localise in time (but complicated)
    - per Ingress interface
    - IP ToS
  - store
    - counters for packets and bytes
    - TCP flags
    - start and stop times
    - a little about routing
- Practicality: aggregate by key
  - flush records using
    - timeout, O(15 seconds), (to separate similar connections, *e.g.*, DNS)
    - when flow record cache is full
    - every X minutes, O(15 minutes), (stop staleness of records)
  - not bi-directional

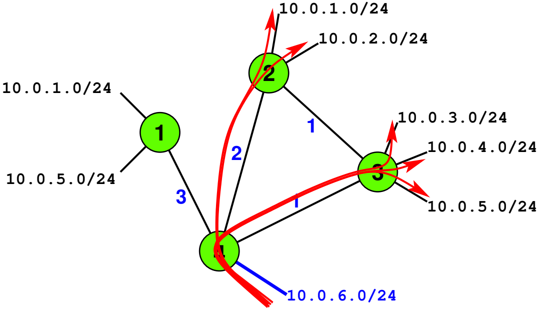
Netflow example application



Example traffic matrix computation

Measured incoming traffic at node 4

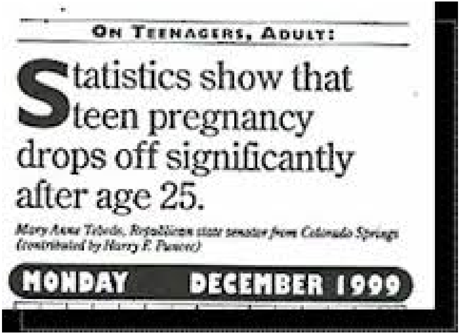
ingress node	source prefix	dest prefix	volume	egress node
4	10.0.6.0/24	10.0.1.0/24	10	2
4	10.0.6.0/24	10.0.2.0/24	11	2
4	10.0.6.0/24	10.0.3.0/24	21	3
4	10.0.6.0/24	10.0.4.0/24	6	3
4	10.0.6.0/24	10.0.5.0/24	3	3



Netflow TM

Netflow can be used to construct a TM

- but you need more data than you think (*e.g.,* topology)
- Netflow isn't universal – historically poor vendor support
- have to sample
  - but almost everyone is hopeless at statistics



- What do the errors in Netflow look like?

Section 3

What do TMs look like?

What do TMs look like?

- A TM really has three dimensions
  - 2 spatial: origin and destination
  - 1 temporal: time of each snapshot

  so we could represent it as a **tensor**
- We usually use a matrix, but it could mean
  - a purely spatial **snapshot** at a particular time
  - a matrix of stacked vector snapshots

$$X = \underbrace{\left[\begin{array}{c|c|c|c} \vdots & \vdots & & \vdots \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_t \\ \vdots & \vdots & & \vdots \end{array}\right]}_{time}$$

- Could have other dimensions
  - traffic types

What do TMs look like?

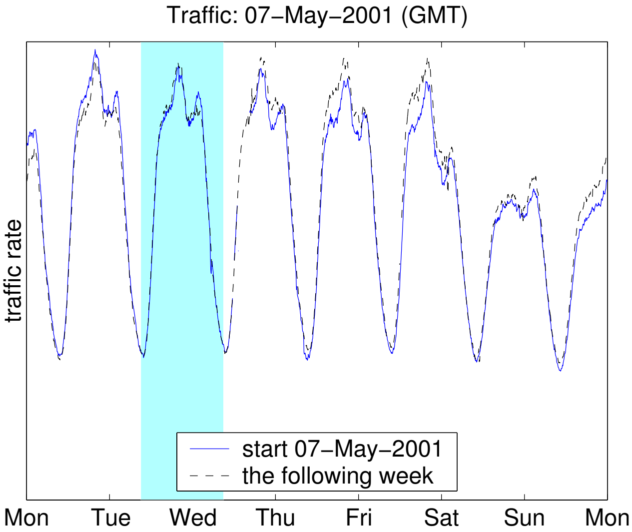
- A TM could contain
  - number of flows
  - number of packets
  - number of bytes

  Mostly they give bytes
- A TM snapshot is usually an average of some time interval
  Common examples are
  - 5 minutes
  - 30 minutes

What do TMs look like?

Temporal patterns

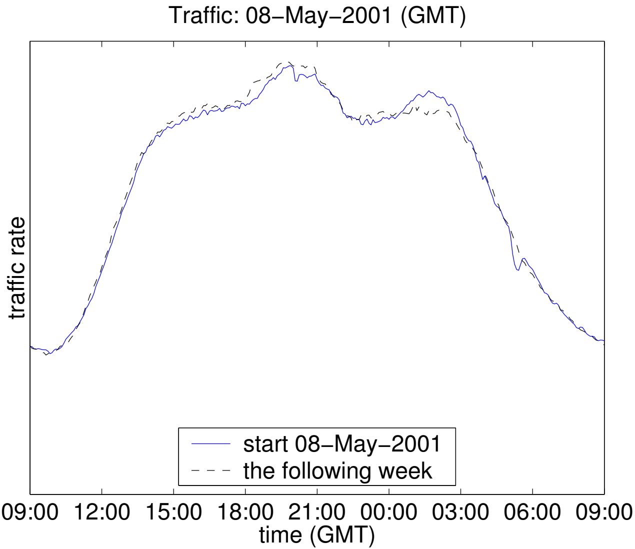
Large ISP [29] local traffic



What do TMs look like?

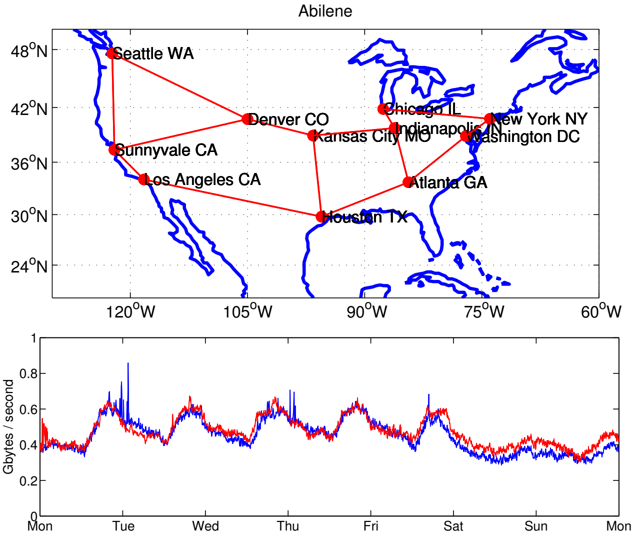
Temporal patterns

Large ISP [29] local traffic

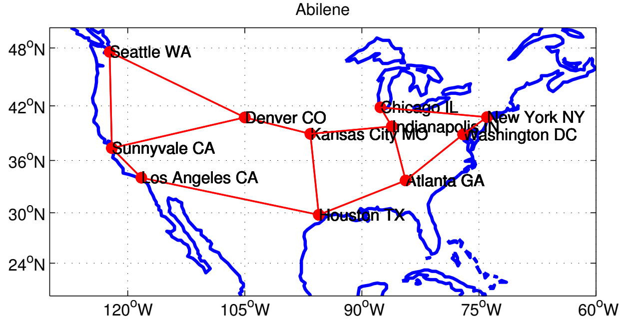


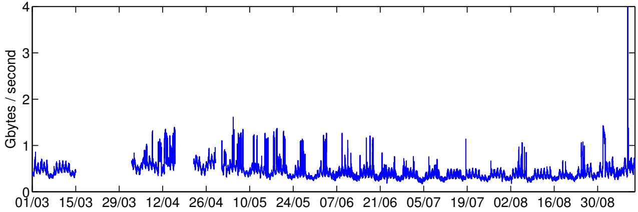
Individuals are random, but the flock is not!

Examples: Abilene c2004



Examples: Abilene c2004





Temporal pattern

$$x(t) = m(t) + \sqrt{am(t)}W(t) + I(t),$$

where

$$m(t) = S(t)\,L(t)$$

and

1. $L(t)$, long-term traffic trend
2. $S(t)$, seasonal (cyclical) component
3. $W(t)$, random (normal) fluctuations
4. $I(t)$, anomaly component
5. a, peakedness

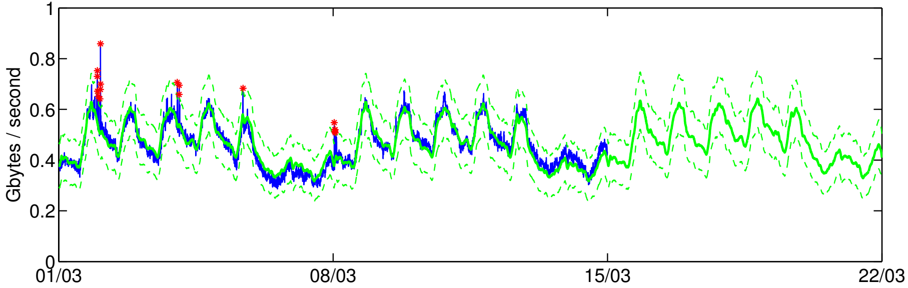
Model rationale

- Period pattern is well known, 24 hours, 1 week
- Multiplexing

$$x_{\text{agg}}(t) = \sum_{i=1}^{K} m_i(t) + \sum_{i=1}^{K} \sqrt{a_i m_i(t)}\, W_i(t) + \sum_{i=1}^{K} I_i(t).$$

  - leads to consistent mean and variance estimates
- Presumption is that growth arises mainly from new sources, not increases in old sources
  - NB: source here might not mean individuals
- It's an easy model to **estimate**

Data and model



What do TMs look like?

Spatial patterns

src	1	2	3	4	5	6	7	8	9	10	11	12	sum
	dst												
1	0.07	0.07	0.43	0.00	0.06	0.12	0.06	0.00	0.05	0.00	0.00	0.25	1.12
2	0.00	4.09	6.42	0.06	7.07	4.42	1.59	0.02	3.24	0.03	0.16	11.09	38.18
3	0.00	4.70	25.48	4.11	13.99	11.53	3.31	87.27	5.22	0.01	0.08	7.70	163.38
4	0.00	1.93	10.25	1.68	5.63	6.11	2.59	0.01	4.11	2.60	0.04	5.92	40.88
5	0.00	4.76	0.25	0.01	24.06	0.04	0.01	0.02	1.24	0.02	0.03	18.05	48.49
6	0.00	2.87	23.73	1.55	13.53	4.78	2.89	0.01	9.45	0.08	0.50	7.64	67.02
7	0.00	0.67	4.79	1.92	3.50	2.24	1.25	0.00	0.93	0.02	0.03	3.31	18.67
8	0.00	4.18	2.58	5.80	26.35	0.17	0.16	1.41	10.88	2.11	3.64	16.67	73.97
9	0.00	8.61	12.34	5.71	18.21	11.05	3.84	0.41	36.36	0.02	0.52	17.31	114.37
10	0.00	0.18	0.04	1.71	1.69	0.00	0.06	5.61	0.96	1.82	8.44	0.36	20.86
11	0.00	3.47	3.28	0.54	8.60	0.13	0.93	3.92	1.77	0.81	0.61	2.32	26.38
12	0.00	18.20	16.04	0.83	34.03	11.18	5.64	0.09	25.57	0.08	0.80	47.02	159.47
sum	0.07	53.74	105.61	23.94	156.73	51.76	22.34	98.77	99.77	7.59	14.84	137.65	772.80

Abilene 5 minute traffic matrix from April 15th, 2004 from 16:25–16:30, in Mbps.

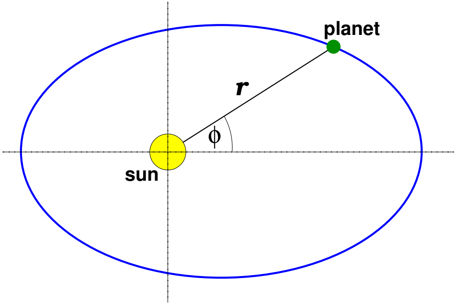
What do TMs look like?

Spatial patterns

- Newtonian gravity

$$F = \frac{GMm}{r^2}$$

  - force depends on mass and distance
  - no dependence on type of mass
    - ★ lead has the same gravitational constant as air



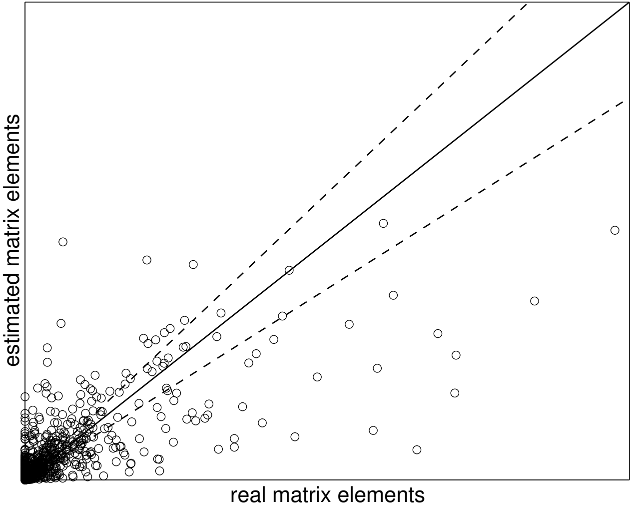
Simple Gravity Model

- Internet traffic model:

$$T(i,j) = \frac{T_{\text{in}}(i) \times T_{\text{out}}(j)}{T_{\text{total}}}$$

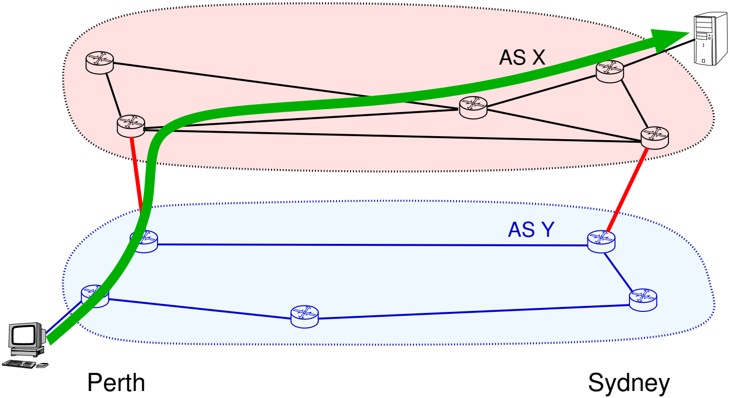
  - traffic between *i* and *j* only depends on how “big” *i* and *j* are
    - no dependence on the type of location
  - different from Newtonian gravity
    - no distance term
  - not a perfect model, but it’s useful

Errors in gravity model [11]



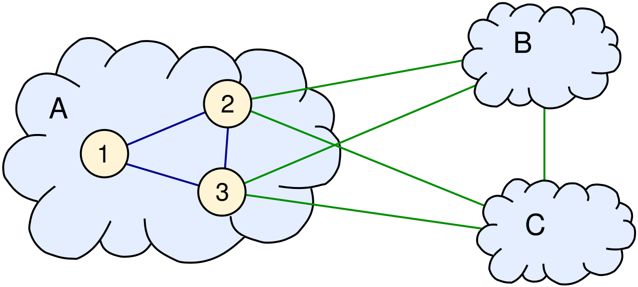
Hot Potato Routing and Gravity Models

- We model OD TMs, but see IE TMs



Generalised Gravity Model

Simple Example with 3 Autonomous Systems

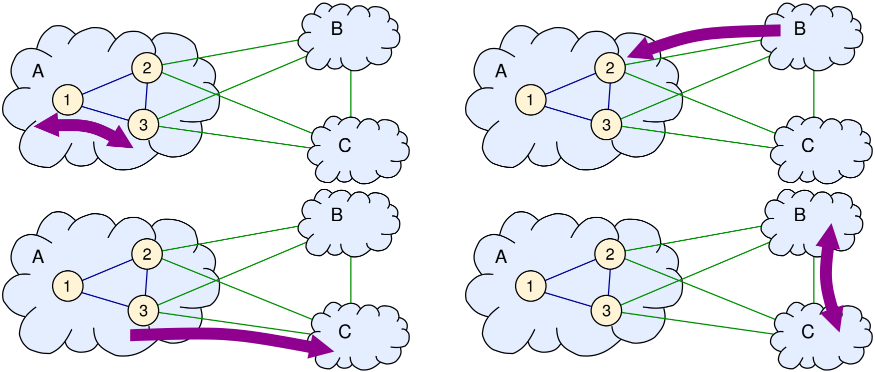


(uniform) gravity model OD traffic matrix

$$X^{(OD)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & B & C \end{array} \\ \left(\begin{array}{ccccc} 1/9 & 1/9 & 1/9 & 1/3 & 1/3 \\ 1/9 & 1/9 & 1/9 & 1/3 & 1/3 \\ 1/9 & 1/9 & 1/9 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 & 1 & 1 \\ 1/3 & 1/3 & 1/3 & 1 & 1 \end{array} \right) \end{array}$$

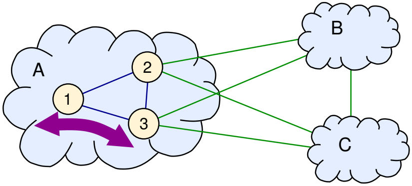
Generalised Gravity Model

There are four classes of flows:



each behaves differently.

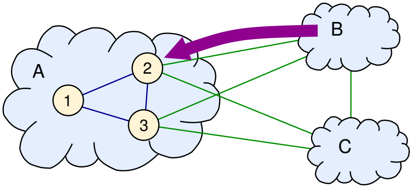
Generalised Gravity Model



We only observe IE TM, which is made up of three components

$$X^{(IE)}_{\text{internal}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \end{array}$$

Generalised Gravity Model

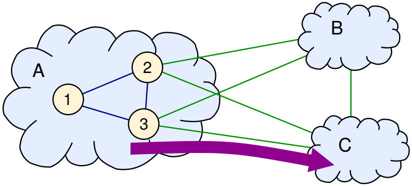


We only observe IE TM, which is made up of three components

$$X^{(IE)}_{\text{arriving}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} \end{array}$$

- assumes traffic from B and C is split evenly over possible entry points (routers 1 and 2)

Generalised Gravity Model



We only observe IE TM, which is made up of three components

$$X^{(IE)}_{\text{departing}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1/3 & 1/3 \\ 0 & 2/3 & 0 \\ 0 & 0 & 2/3 \end{pmatrix} \end{array}$$

- assumes hot potato routing
- internal IGP weights are equal

Generalised Gravity Model

Total IE traffic matrix

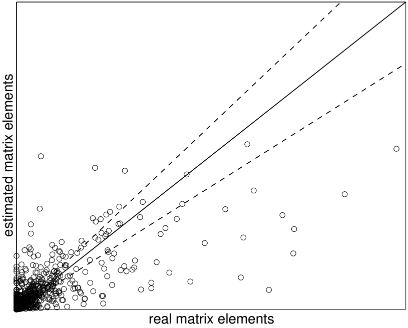
$$X^{(IE)} = \begin{pmatrix} 1/9 & 4/9 & 4/9 \\ 4/9 & 10/9 & 4/9 \\ 4/9 & 4/9 & 10/9 \end{pmatrix}$$

which is far from fitting the gravity model,

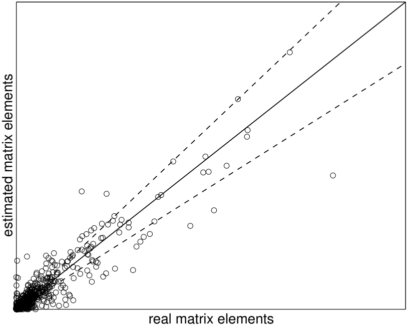
$$X^{(IE)}_{\text{gravity}} = \begin{pmatrix} 1/5 & 2/5 & 2/5 \\ 2/5 & 4/5 & 4/5 \\ 2/5 & 4/5 & 4/5 \end{pmatrix}$$

even though all of its OD components do fit the gravity model

Generalised Gravity Model Errors



Gravity model



Generalised Gravity Model

In general

There are lots of complexities not included in the gravity model

- IE matrices – not symmetric
- Diagonal entries are always a problem
- People aren't sheep

- new(ish) tech: CDNs, clouds, ...

We could start down a long road of modelling here, which I don't want to do just yet, but note that tomographic techniques fix some of the errors using link data.

In general

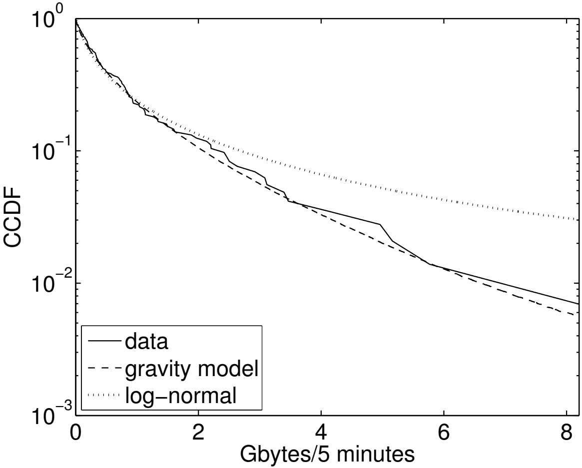
There are lots of complexities not included in the gravity model

- IE matrices – not symmetric
- Diagonal entries are always a problem
- People aren't sheep
  - Australians aren't New Zealanders
- new(ish) tech: CDNs, clouds, ...

We could start down a long road of modelling here, which I don't want to do just yet, but note that tomographic techniques fix some of the errors using link data.

Distributional properties [27]

TM entries are not heavy tailed



NB: here the gravity model is formed from row/col sums that are drawn from an exponential distribution (more on this later)

Section 4

How do you use a TM?

Network Management

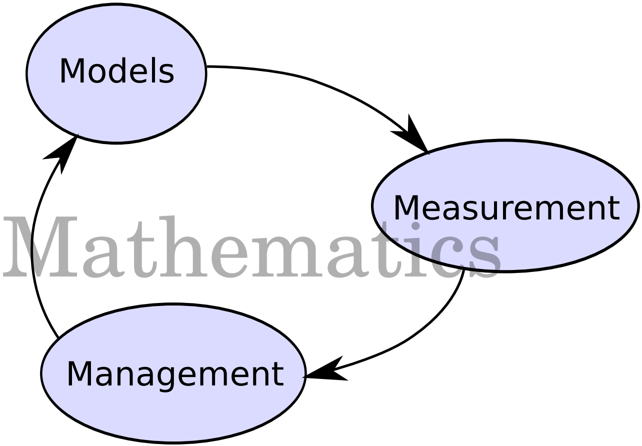
Network management, as defined by the OSI [30] FCAPS

- F Fault – recognise, isolate, correct, prevent faults
- C Configuration – programming a set of flexible devices (switches, routers, and servers) to implement the high-level goals of the network operator
- A Accounting – gather usage statistics of users primarily for billing
- P **Performance** – ensure network performance remains at “acceptable” levels
- S Security – ensure availability, integrity, confidentiality

But also faults, and accounting ...

NM^4

Network Management is an **integrated process** not a set of tasks



Network engineering goals

- Reliability

Network engineering goals

- Reliability
- Reliability

Network engineering goals

- Reliability
- Reliability
- Cost

Network engineering goals

- Reliability
- Reliability
- Cost
- Performance

Network engineering goals

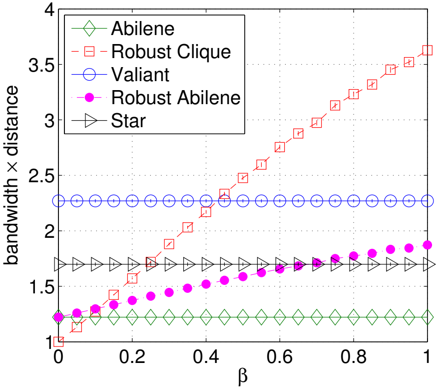
- Reliability
- Reliability
- Cost
- Performance
- Reliability

Network Reliability Analysis

- Answer “what if?” questions
  - what if link X fails?
- It’s not just about connectivity
  - rerouted traffic can cause congestion
- To do this we need
  - network configuration
  - fault risks
  - traffic data
  - performance models
- An example

Some interesting bits

All TMs have errors – how does that affect answers?



Some methods (of network design) are highly-sensitive to errors, and others aren't!!! [31] (2014)

- analysis required ability to generate variations around a TM

Other Applications for Network Operators

Usually involve prediction of TMs, though over different **horizons**

- Network planning
  - 6 months to a year: planning capacity
  - 1 day to 2 weeks: traffic engineering
- Detecting unusual traffic (anomaly detection)
  - minutes to hours

Synthesis – the next challenge

- Network operators design based on “real” TMs
  - now there are various methods to get the required data
  - need to be able to work with errors
  - synthesis can help [31]
- Researchers need data as well
  - but network operators don’t release TM data
  - even if they did, they would never release enough
    - *e.g.,* to do stats on results
  - even if they did provide enough, researchers need control
    - *e.g.,* to extrapolate results

So where do we (the research community) get TM data?

Section 5

What do I do if I don't have any data?

Pop quiz

If you choose an answer to this question at random, what is the chance you will be correct?

- A 25%
- B 50%
- C 66%
- D 25%

Specific Applications for Researchers

Usually involve an *ensemble* of traffic matrices

- Designing a new
  - routing protocol
- Testing algorithms for
  - anomaly detection
  - traffic engineering or network planning
- Synthesising networks
  - traffic is a fundamental input [32, 33]

Could also apply for green-fields planning

Data is hard to get

- Network operators don't share
  - traffic data is proprietary
  - traffic data is private
- How representative is any set anyway?
  - Abilene might be thought outdated
- We need lots of data for some tasks
  - *e.g.,* anomaly detection needs to estimate small probabilities [34]
  - more than you get from one network
- We might need data where there is no network
  - green-fields planning
  - what happens when my network scales up $\times$ 10?

Synthesis saves the day!

Reproducible research

> An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the Figures.
>
> *Buckheit and Donoho [35]*

- Some Internet data can never be shared
- Too much Internet research is NOT reproducible
  - this stifles science
  - it results (sometimes) in incorrect results
  - it encourages fraud and other scientific malfeasance
- Synthesis provides a (partial) solution

Synthesis Requirements: SCERC

- **Simplicity:**
  - Occam's razor
  - Principle of parsimony
  - Bonini's paradox

> Everything simple is false.
> Everything which is complex is unusable.
> *Paul Valéry*

- **Control:** test methods against assumptions.
- **Efficiency:** TMs can be big, plus we want to generate many.
- **Realism:** simplest to think you understand, hardest to really understand!
- **Consistency:** allow apples to apples comparisons

Synthesis formalities

- We want to generate an **ensemble**
  - collection of **instances** with some probability measure
  - need to have controlled statistical variation - there is no point in making all instances the same!
- Want to incorporate some knowledge or assumptions
  - maybe because we have some data
  - maybe because we want to compare our results to someone elses'
- Don't want extraneous, unstated assumptions

The answer is synthesis (or simulation)

- The question is, using what model?
- I have a few answers, and they go in the order

$$\text{Simple} \\ \Downarrow \\ \text{Complex} \\ \Downarrow \\ \text{Simple}$$

Simple again

- What if we started with a set of "axioms"
  - things we know about a set of traffic matrices
  - ensemble properties
- How would we build models that
  - include the parts we want
  - don't accidentally include other assumptions

Simple again

- What if we started with a set of "axioms"
  - things we know about a set of traffic matrices
  - ensemble properties
- How would we build models that
  - include the parts we want
  - don't accidentally include other assumptions
- Maximum entropy [36]
  - Maximum entropy $\Rightarrow$ gravity-like models
  - We have code `https://github.com/ptuls/MaxEntTM`

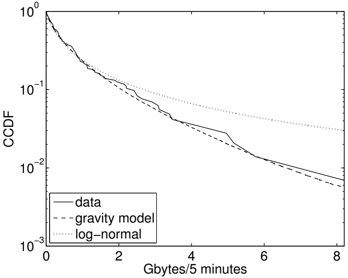
Simple

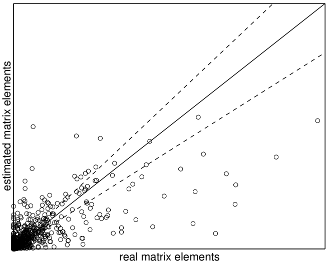
Use the gravity model [27]

1. Generate random row and column sums
   - exponential random variables seemed to work
2. Calculate the gravity model
   - it's just multiplication
3. Possibly scale to match required total

Simple

- Pros
  - very simple
  - matches distribution well
- Cons
  - structure isn't right
  - lack of control





Complex

You can think of any number of ways to include more complex models, ideas, assumptions, ...

Challenges

- Loose simplicity
- Loose efficiency
- Testing realism
- In theory you gain control, but in practice you often end up with many parameters which are hard to estimate (from data), or guess by other means

Simple again

- What if we started with a set of “axioms”
  - things we know about a set of traffic matrices
  - ensemble properties
- How would we build models that
  - include the parts we want
  - don’t accidently include other assumptions

Simple again

- What if we started with a set of “axioms”
  - things we know about a set of traffic matrices
  - ensemble properties
- How would we build models that
  - include the parts we want
  - don’t accidently include other assumptions
- **Maximum entropy** does this [36]

Maximum Entropy Idea [37]

- Shannon entropy

$$H(X) = -\sum_{x} p(x) \log p(x),$$

  can be seen as a measure of how much information we need to describe X
- Another way to say that is it's a measure of uncertainty
- If we find a distribution of X that maximises $H(X)$ subject to any constraints, it must be the one that imposes the least possible *a priori* assumptions or knowledge on X
- Find $p(x)$ is just an optimisation problem

Simple Case

Imagine we knew certain features of the data

$$\mathbb{E}\Big[\sum_j X_{i,j}\Big] = \boldsymbol{r} \textbf{ (outgoing)}$$

$$\mathbb{E}\Big[\sum_i X_{i,j}\Big] = \boldsymbol{c} \textbf{ (incoming)}$$

$$\mathbb{E}\Big[\sum_{i,j} X_{i,j}\Big] = T \textbf{ (total)}$$

$$\sum_i r_i = \sum_j c_j = T \textbf{ (consistency)}$$

Then the natural MaxEnt model is a gravity model

$$X = T \underbrace{U}_{\text{row}} \underbrace{V^T}_{\text{column}}$$

where U and V are vectors of independent exponential random variables whose average matches the row and col sums.

This is (almost) the gravity model proposed earlier!

More complex cases

- Spatio-temporal structure
- Constraints on variance (*e.g.,* errors in measurements)
- Soft v hard constraints
- Convex constraints

Works in a very modular, building-block manner

Finding the maximum entropy distribution $\neq$ sampling from it

- Simple cases have closed forms, *i.e.,* are easy
- More complex cases we need to use a sampling algorithm
  - *e.g.,* MCMC (Markov Chain Monte Carlo)
- These aren't always tractable without some care!
  - we have reasonable code for common TM cases
    `https://github.com/ptuls/MaxEntTM`

Other plusses

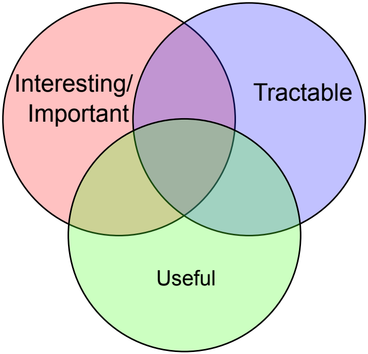
- Maximum entropy creates a matrix between
  - assumptions
  - models
- We see this with gravity model
  - now we know why it is a good model to start with, and when it is good, and when it is bad
- (truncated) normal implies mean and variance

Recap

1. Intro
2. How do you get a TM?
3. What do TMs look like?
4. How do you use a TM?
5. What do I do if I don't have any data?

Conclusion

- network performance shouldn't be considered without thinking about the data/measurements we can have, and the tasks at hand



- connecting research to real problems is
  - necessary if you want to have impact
  - rewarding, because useful problems are often interesting

P. Tune and M. Roughan, "Internet traffic matrices: A primer," in *Recent Advances in Networking, Vol. 1* (H. Haddadi and O. Bonaventure, eds.), ACM SIGCOMM, August 2013.

J. Kruithof, "Telefoonverkeersrekening," *De Ingenieur*, vol. 52, no. 8, pp. 15–25, 1937.

D. Bear, "Some theories of telephone traffic distribution: A critical survey," in *International Teletraffic Congress (ITC) 7*, (Stockholm), pp. 531/1–531/5, 1973.
http://www.i-teletraffic.org/bibliographic-database/itc-7/.

R. B. Potts and R. M. Oliver, *Flows in Transportation Networks*.
Academic Press, 1972.

Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data," *J. Am. Statist. Assoc.*, vol. 91, no. 433, pp. 365–377, 1996.

C. Tebaldi and M. West, "Bayesian inference on network traffic," *Journal of the American Statistical Association*, June 1998.

J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography," *Journal of the American Statistical Association*, pp. 1063–1075, December 2000.

A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," in *Proceedings ACM SIGCOMM'00*, August/September 2000.

A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "Netscope: Traffic engineering for IP networks," *IEEE Network Magazine*, pp. 11–19, March/April 2000.

A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *ACM SIGCOMM*, (Pittsburg, USA), August 2002.

Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *ACM SIGMETRICS*, (San Diego, California), pp. 206–217, June 2003.

- Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *ACM SIGCOMM*, (Karlsruhe, Germany), pp. 301–312, August 2003.
- A. Nucci, R. Cruz, N. Taft, and C. Diot, "Design of IGP link weights for estimation of traffic matrices," in *IEEE Infocom*, (Hong Kong), March 2004.
- A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large IP backbone – a comparison on real data," in *ACM/Usenix Internet Measurement Conference*, (Taormina, Sicily, Italy), October 2004.
- Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information: making use of multiple data sources," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 133–144, 2006.
- T. Telkamp and S. Schnitter, "Best practices for determining the traffic matrix in IP networks," in *Apricot*, (Kyoto, Japan), February 2005.
- V. Erramilli, M. Crovella, and N. Taft, "An independent-connection model for traffic matrices," in *ACM SIGCOMM Internet Measurement Conference (IMC06)*, (New York, NY, USA), pp. 251–256, ACM, 2006.
- G. Liang, N. Taft, and B. Yu, "A fast lightweight approach to origin-destination IP traffic estimation using partial measurements," *Special joint issue of IEEE/ACM Transaction on Networking and IEEE Transactions on Information Theory*, June 2006.
- 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 *ACM SIGMETRICS*, pp. 362–373, 2005.
- Y. Zhang, M. Roughan, W. Willinger, and L. Qui, "Spatio-temporal compressive sensing and Internet traffic matrices," in *ACM Sigcomm*, (Barcelona, Spain), pp. 267–278, August 2009.
- V. Bharti, P. Kankar, L. Setia, G. Gürsun, A. Lakhina, and M. Crovella, "Inferring invisible traffic," in *Proceedings of the 6th International Conference*, Co-NEXT '10, (New York, NY, USA), pp. 22:1–22:12, ACM, 2010.
- A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *ACM SIGMETRICS / Performance*, 2004.

A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM*, 2004.

A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *ACM SIGCOMM Internet Measurement Conference*, (Taormina, Sicily, Italy), October 2004.

Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 947–960, October 2005.

A. Nucci, A. Sridharan, and N. Taft, "The problem of synthetically generating IP traffic matrices: Initial recommendations," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 3, 2005.

M. Roughan, "Simplifying the synthesis of Internet traffic matrices," *ACM SIGCOMM Computer Communications Review*, vol. 35, pp. 93–96, October 2005.

M. Roughan, "A case-study of the accuracy of SNMP measurements," *Journal of Electrical and Computer Engineering*, vol. 2010, 2010.
Article ID 812979, doi:10.1155/2010/812979. http://www.hindawi.com/journals/jece/2010/812979.html.

M. Roughan, *Guide to Reliable Internet Services and Applications*, ch. Robust Network Planning.
Springer, 2010.

"Information processing systems – open systems interconnection – basic reference model – part 4: Management framework." ISO/IEC 7498-4.

P. Tune and M. Roughan, "Network-design sensitivity analysis," in *ACM SIGMETRICS*, June 2014.

R. Bowden, M. Roughan, and N. Bean, "Cold: PoP-level topology synthesis," in *CoNext*, (Sydney, Australia), December 2014.

L. Li, D. Alderson, W. Willinger, and J. Doyle, "A first-principles approach to understanding the Internet's router-level topology," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, (New York, NY, USA), pp. 3–14, ACM, 2004.

H. Ringberg, M. Roughan, and J. Rexford, "The need for simulation in evaluating anomaly detectors," *ACM SIGCOMM CCR*, vol. 38, pp. 55–59, January 2008.

J. B. Buckheit and D. L. Donoho, *Wavelets and Statistics*, vol. 103, ch. WaveLab and Reproducible Research, pp. 55–81. Springer: Lecture Notes in Statistics, 1995.

P. Tune and M. Roughan, "Spatiotemporal traffic matrix synthesis," in *ACM Sigcomm*, (London, UK), August 2015.

E. Jaynes, "Information theory and statistical methanics," *Physical Review*, vol. 106, pp. 620–630, May 1957.

Bonus frames