Traffic Matrix Estimation

EL 933, Class 8
Yong Liu
11/01/2005

Motivation

- Traffic Matrix (TM): the volume of traffic between any two points of the network
- TM is important for:
  - load balancing
  - failure recovery
  - resource provisioning
  - network planning
- TM is difficult to obtain
  - too expensive to measure directly
  - not straightforward to infer from indirect measurement

Papers Today

Outline

- Existing Approaches:
  - Linear Programming
  - Bayesian
  - Expectation Maximization

- Evaluation
  - accuracy/sensitivities
  - model assumptions

- New Directions
  - incorporate more network information
  - preliminary validations

Traffic Matrix (TM)

- Traffic volume between ingress-egress (OD) pairs
- Ingress and egress points can be routers or POPs (Point of Presence, typically a metropolitan area).

IP Backbone: POP-to-POP view

Direct TM Measurement

- TM can be computed directly by collecting flow-level data at ingress points
  - additional features on all routers (Netflow)
  - reduced forwarding performance at routers.
  - large amount of traces

- Too expensive to have!
TM Estimation

- Available information:
  - link counts from SNMP data.
  - routing information. (Weights of links)

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 1 \\
1 & 1 & 0 \\
0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
X_3
\end{bmatrix}
\]

TM Estimation: difficulty

- link rate vector is determined by TM
  - routing matrix \( A(i,j)=1 \) if ingress-egress pair \( j \) traverses link \( i \).
  - \( Y \): link data vector, \( y(i) \) data rate on link \( i \);
  - \( X \): TM vector, \( X(j) \) traffic rate for OD pair \( j \)
  - \( Y = AX \)
- TM is not uniquely determined by link rate vector
  - more \( X \) than \( Y \)
  - \( A \) is generally not invertible
  - one measurement data \( \rightarrow \) infinite many TMs
  - the estimation problem is highly underdetermined

Possible Solutions

- Issue: which TM is the most likely one?
- Techniques:
  - deterministic: introduce new objective and/or constraints
    - Linear Programming (LP)
  - statistical: assumptions on TM distribution
    - Bayesian estimation
    - Expectation Maximization
  - incorporate new information by exploring network structures

Scheme 1: Linear Programming

- Objective
  - maximize the total amount of traffic routed through the network
    \[ \max \sum_{j=1}^{n} y_j x_j \]
- Constraints
  - link count:
    \[ \sum_{i=1}^{n} a_{ij} x_i < y_j \]
  - flow conservation:
    \[ \sum_{i=\text{src}(k)}^{n} y_{i;k} - \sum_{i=\text{dst}(k)}^{n} y_{i;k} = \begin{cases} x_k & \text{if } j=\text{src}(k) \\ -x_k & \text{if } i=\text{dst}(k) \\ 0 & \text{otherwise} \end{cases} \]
- Observation
  - problem: large value for OD pairs with small number of hops
  - solution: put more weight on pairs separated by greater distances
Statistical Approaches

1. Assumption on elements of TM
2. Prior TM
3. ISIS Weights
4. SNMP Counts

- Estimate parameters of distributions (Conditioned on observed link counts)
- Compute means of distributions
- Iterative Proportional Fitting

General Diagram for statistical approach

Scheme 2: Bayesian Approach

- Bayesian Rule: $p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)}$
- Goal: given $Y$, compute posterior prob. dist. $p(X|Y)$, under prior dist. of $X$
- Assumption: OD pairs are independent
- Poisson with unknown rates $\Lambda$
- Both $X$ and $\Lambda$ need to be estimated

Simulation method: Markov Chain Monte Carlo (MCMC) to iteratively obtain the posterior
1. Start with a prior matrix $X^0$
2. Draw value of $\Lambda^i$ from $p(\Lambda|X^i, Y)$
3. Using this $\Lambda^i$, draw a value of $X^{i+1}$ from $p(X|\Lambda^i, Y)$
4. Iterate until feasible solution is found
- Reduce problem size by exploring network structure
  $X_1 = A_1^{-1}(Y - A_2X_2)$

Scheme 3: Expectation Maximization

- Assumes OD pairs are ind. dist. Gaussian.
  $X \sim \text{Normal}(\Lambda, \Xi)$
- $Y=AX$ implies link count are still Gaussian
  $Y \sim \text{Normal}(AA\Lambda, A\Xi A')$
- Requires a prior for initialization.
- Incorporates multiple sets of link measurements.
- Uses EM algorithm to compute MLE.
### E-M Algorithm

- **MLE:** \( L(\theta \mid Y) = p(Y \mid \theta) \), want \( \theta^* = \text{argmax} \ L(\theta \mid Y) \)
- \( Y \) observed, call \( Y \) incomplete data
- \( Z = (X, Y) \) complete data, \( X \) is not directly observable
- \( L(\theta \mid Z) = p(z \mid \theta) = p(x, y \mid \theta) = p(x \mid y, \theta) \ p(y \mid \theta) \)
- Iterative EM algorithm
  - **E-step:** find expected value of the complete-data log-likelihood \( \log p(X, Y \mid \Theta) \) w.r.t the unknown data \( X \) given observation \( Y \) and current parameter estimates
  - **M-step:** maximize expectation computed in E-step

### E-M algorithm

- **E-Step**
  \[
  Q(\Theta, \Theta^{(i-1)}) = E[\log p(X, Y \mid \Theta) | Y, \Theta^{(i-1)}] = \int_{x \in \Omega} \log p(x, Y \mid \Theta) f(x | Y, \Theta^{(i-1)}) dx
  \]
- **M-Step**
  \[
  \Theta^{(i)} = \text{argmax}_{\Theta} Q(\Theta, \Theta^{(i-1)})
  \]

### Evaluation Method

- Difficult to obtain traffic matrix directly from operational networks
- Evaluate through simulations
  - two topologies: 4 nodes, 14 nodes PoP topology
  - hypothetical link weights: proprietary
  - synthetic TM: Constant, Poisson, Gaussian, Uniform and Bimodal
- **Comparison criteria:**
  - estimation errors.
  - sensitivity to initial dist.
  - sensitivity to distribution assumptions

### Results: Linear programming vs. Statistical Methods

- **Linear Programming method performs poorly**
  - Problem: tries to match OD pairs to link counts
  - Different objective functions give similar results
  - Too high error for use in practical networks
- **Bayesian and EM**
  - EM beats Bayesian in terms of average error and worst case error (MCMC algorithm is stochastic)
  - Estimation errors correlated to heavily shared links (links with many OD flows are more likely to be misestimated)
**4-node topology**

Figure 3: 14-NODE Tier-I POP Topology

**4-node topology results**

<table>
<thead>
<tr>
<th>Original TM</th>
<th>Estimated TM</th>
<th>Error%</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayesian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| AR 318     | 318          | 0      |
| LP         | 318          | 0      |
| Bayesian    | 318          | 0      |
| EM         | 318          | 0      |

Table 1: Estimates for 4-node Topology

**Results: Goodness of prior**

<table>
<thead>
<tr>
<th>Dist</th>
<th>Avg</th>
<th>Max</th>
<th>.2</th>
<th>.5</th>
<th>.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>0.20</td>
<td>1.16</td>
<td>0.60</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.26</td>
<td>2.31</td>
<td>0.58</td>
<td>0.83</td>
<td>0.91</td>
</tr>
<tr>
<td>Poisson</td>
<td>0.23</td>
<td>1.99</td>
<td>0.57</td>
<td>0.90</td>
<td>0.94</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.23</td>
<td>1.78</td>
<td>0.59</td>
<td>0.88</td>
<td>0.94</td>
</tr>
<tr>
<td>Bi-modal</td>
<td>0.41</td>
<td>5.00</td>
<td>0.41</td>
<td>0.76</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Table 3: Bayesian, 14-node topology, Good prior.

**Results: Goodness of prior**

<table>
<thead>
<tr>
<th>Dist</th>
<th>Avg</th>
<th>Max</th>
<th>.2</th>
<th>.5</th>
<th>.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>0.12</td>
<td>0.54</td>
<td>0.81</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.13</td>
<td>1.07</td>
<td>0.75</td>
<td>0.96</td>
<td>0.98</td>
</tr>
<tr>
<td>Poisson</td>
<td>0.11</td>
<td>0.42</td>
<td>0.31</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.14</td>
<td>0.57</td>
<td>0.71</td>
<td>0.98</td>
<td>1.00</td>
</tr>
<tr>
<td>Bi-modal</td>
<td>0.22</td>
<td>0.90</td>
<td>0.82</td>
<td>0.89</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 5: EM, 14-node topology, Good prior.

**Results: Goodness of prior**

<table>
<thead>
<tr>
<th>Dist</th>
<th>Avg</th>
<th>Max</th>
<th>.2</th>
<th>.5</th>
<th>.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>0.41</td>
<td>2.12</td>
<td>0.41</td>
<td>0.75</td>
<td>0.87</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.43</td>
<td>5.55</td>
<td>0.43</td>
<td>0.71</td>
<td>0.84</td>
</tr>
<tr>
<td>Poisson</td>
<td>0.37</td>
<td>2.83</td>
<td>0.38</td>
<td>0.76</td>
<td>0.86</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.41</td>
<td>5.26</td>
<td>0.41</td>
<td>0.79</td>
<td>0.89</td>
</tr>
<tr>
<td>Bi-modal</td>
<td>0.63</td>
<td>4.97</td>
<td>0.29</td>
<td>0.56</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 4: Bayesian, 14-node topology, Bad prior.

**Results: Goodness of prior**

<table>
<thead>
<tr>
<th>Dist</th>
<th>Avg</th>
<th>Max</th>
<th>.2</th>
<th>.5</th>
<th>.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>0.22</td>
<td>1.00</td>
<td>0.53</td>
<td>0.90</td>
<td>0.96</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.24</td>
<td>1.01</td>
<td>0.57</td>
<td>0.85</td>
<td>0.92</td>
</tr>
<tr>
<td>Poisson</td>
<td>0.23</td>
<td>1.28</td>
<td>0.55</td>
<td>0.88</td>
<td>0.95</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.24</td>
<td>1.11</td>
<td>0.48</td>
<td>0.86</td>
<td>0.95</td>
</tr>
<tr>
<td>Bi-modal</td>
<td>0.39</td>
<td>1.50</td>
<td>0.42</td>
<td>0.66</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 6: EM, 14-node topology, Bad prior.
Results: Goodness of prior

- Goodness of prior matrix (seed values)
  - Bayesian more sensitive to the prior matrix than EM
  - Possible reason: Bayesian has stochastic convergence while EM method has deterministic convergence behavior (analytic, guaranteed improvement)
- After a certain point, additional measurements don’t provide additional gain
  - The time-varying extension provides only a small improvement

Results: Marginal gains

- How much better we can do if we can directly measure some components of the TM?
  - Operators can afford direct measurement on some routers
- Three ways to add rows:
  - randomly,
  - row-sum(by traffic volume),
  - error magnitude

Other results:

- Which OD pairs are most difficult to estimate?
  - error rate increases as the link-sharing factor
  - error rate increases also as path length
- Validation of model assumptions?
  - Poisson and Gaussian assumption holds well but only for certain hours during the day

- Error rate drops off roughly linearly with each additional row added
- Bayesian not sensitive to order rows are added
- EM does better when rows added by largest-error first
- Reduction in adding a row is 2% for 13OD pairs
New Directions

- Lessons learned:
  - Model assumptions do not reflect the true nature of traffic. (multimodal behavior)
  - Dependence on priors
  - Link count is not sufficient (Generally more data is available to network operators.)

- Proposed Solutions:
  - Use choice models to incorporate additional information.
  - Generate a good prior solution.

PoP Fanout

- What is the fraction of outbound traffic of PoP i going to PoP j?
  - \( X_{ij} = O_i \alpha_{ij} \)
  - Estimation of Fanout: \( \{\alpha_{ij}\} \)

- Solution via Discrete Choice Models (DCM).
  - User choices: traffic demand
  - ISP choices: routing choice

Choice Models

- Decision makers: PoPs
- Set of alternatives: egress PoPs.
- Attributes of decision makers and alternatives: attractiveness (capacity, number of attached customers, peering links).
- Utility maximization with random utility models.

Random Utility Model

- Utility of PoP i choosing to send packet to PoP j:
  - \( U_j^i = V_j^i + \epsilon_j^i \)
- Choice problem:
  - \( P_C^i(j) = P[U_j^i = \max_{k \in C}\{U_k^i\}] \)
- Deterministic component:
  - \( V_j^i(X_j^i) = \sum_{m=1}^{M} \mu_m w_j^i(m) + \gamma_j \)
- Random component: mlogit model used.
  - \( X_{ij} = O_i \frac{e^{V_j^i}}{\sum_{k \in C} e^{V_k^i}} \)
Results

- Two different models
  - Model 1: attractiveness
  - Model 2: attractiveness + repulsion

Summary

- Existing TM estimation approaches are not very accurate
  - path length
  - link sharing factors
  - prior distribution
- Model assumptions are not always validated
- Propose new models to exploit more network information
- Open question
  - How to generate the good priors
  - How to efficiently incorporate the POP features

Outline

- Background
- Tomogravity method
  - simple gravity
  - generalized gravity
  - tomogravity
- Performance evaluation with real traffic matrices
ISP Network View

- Customer & Peers
- Access links
- Peering links
- Four traffic categories
  - Transit
  - Outbound
  - Inbound
  - Internal

Tomogravity = Gravity + Tomography

- Two step modeling.
  - **Gravity Model**: Initial solution obtained using edge link load data and ISP routing policy.
  - **Tomographic Estimation**: Initial solution is refined by applying quadratic programming to minimize distance to initial solution subject to tomographic constraints (link counts).

Gravity Modeling

- General formula: \( X_{ij} = \frac{R_i \cdot A_j}{f_{ij}} \)
  - Newton’s gravity law
  - Movement of people, goods, info. between areas
- Simple gravity model: traffic between edge links.
  \[
  T(l_i, l_j) = \frac{T_{\text{in}}(l_i) \cdot T_{\text{out}}(l_j)}{\sum_k T_{\text{in}}(l_k) \cdot T_{\text{out}}(l_k)},
  \]
  \[
  T(l_i, l_j) = \frac{T_{\text{in}}(l_i) \cdot T_{\text{out}}(l_j)}{\sum_k T_{\text{in}}(l_k) \cdot T_{\text{out}}(l_k)}.
  \]

Generalized Gravity Model

- incorporate more info. from link classification and routing policy

Under these assumptions the outbound traffic from access link \( a_i \in A \) to peering link \( p_m \in P \) is

\[
T_{\text{outbound}}(a_i, p_m) = \begin{cases} 
\frac{\sum_{a_k \in A} T_{\text{in}}(a_k) \cdot T_{\text{out}}(a_k)}{\sum_{a_k \in A} T_{\text{in}}(a_k)} & \text{if } p_m = X(a_i, P_j), \\
0, & \text{otherwise}.
\end{cases}
\]

The inbound traffic from peering link \( p_i \) to access link \( a_j \) is

\[
T_{\text{inbound}}(p_i, a_j) = T_{\text{in}}(p_i) \cdot \frac{\sum_{a_k \in A} T_{\text{in}}(a_k)}{\sum_{a_k \in A} T_{\text{in}}(a_k)}.
\]

external traffic models
Generalized Gravity Model

The internal traffic from access link $a_i$ to access link $a_j$ is

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

where

$$T_{\text{internal}}^\text{out}(a_j) = T_{\text{link}}^\text{out}(a_j) - \sum_{p_k \in P} T_{\text{link}}^\text{inbound}(p_k, a_j)$$

$$= T_{\text{link}}^\text{out}(a_j) \left( 1 - \sum_{p_k \in P} \frac{T_{\text{link}}^\text{in}(p_k)}{\sum_{a_k \in A} T_{\text{link}}^\text{in}(a_k)} \right).$$

Tomographic Approach

- Solution need to satisfy the link constraints

1. route
2. route
3. route

\[ Y = AX \]

New Solution: Tomo-gravity

- Under-constrained linear inverse problem
- Find additional constraints based on models
  - Typical approach: use higher order statistics
- Disadvantages
  - Complex algorithm – doesn’t scale
    - Large networks have 1000+ nodes, 10000+ routes
  - Reliance on higher order statistics is not robust given the problems in SNMP data
    - Artifacts, Missing data
    - Violations of model assumptions (e.g. non-stationarity)
    - Relatively low sampling frequency: 1 sample every 5 min
    - Unevenly spaced sample points
  - Not very accurate at least on simulated TM

- “Tomo-gravity” = tomography + gravity modeling
- Exploit topological equivalence to reduce problem size
  - One pair of BRs responsible for traffic between two PoP
  - Eliminate empty demands reduce unknowns by factor of 10
- Use least-squares method to get the solution, which
  - Satisfies the constraints
  - Is closest to the gravity model solution
  - Can use weighted least-squares to make more robust
Quadratic Programming: illustration

**QP formulation:** \[ \text{min} ||X - X_g|| \]
subject to \[||AX - Y||\] is minimized

![Graph showing least square solution, gravity model solution, and constraint subspace]

How to Validate?

- **Simulate and compare**
  - Problems
    - How to generate realistic traffic matrices
    - Danger of generating exactly what you put in
  - Measure and compare
    - Problems:
      - Hard to get Netflow (detailed direct measurements) along whole edge of network
      - If we had this, then we wouldn’t need SNMP approach
      - Actually pretty hard to match up data
      - Is the problem in your data: SNMP, Netflow, routing, ...
  - Our method
    - Novel method for using partial, incomplete Netflow data

Validation Method

- **Use partial, incomplete Netflow data**
  1. **Measure partial traffic matrix** \(X_p\)
     - Netflow covers 70+\% traffic
  2. **Simulate link loads** \(Y_p = A^T X_p\)
     - \(Y_p\) won’t match real SNMP link loads
  3. **Solve** \(Y_p = A^T X\)
  4. **Compare** \(X\) with \(X_p\)

- **Advantage**
  - Realistic network, routing, and traffic
  - Comparison is direct, we know errors are due to algorithm not errors in the data
  - Can test robustness by adding noise to \(Y_p\)

Naïve Approach: raw least square

- **In real networks the problem is highly under-constrained**
Simple Gravity Model

Better than naïve, but still not very accurate

Generalized Gravity Model

Fairly accurate given that no link constraint is used

Tomo-gravity: Accuracy

Accurate within 10-20% (esp. for large elements)

Comparison

\[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2} \]

\[ RMSRE(T') = \sqrt{\frac{1}{N'} \sum_{i=T}^{N} \left( \frac{\hat{x}_i - x_i}{x_i} \right)^2} \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>LSE weights</th>
<th>traffic matrix errors</th>
<th>core link errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial solution</td>
<td></td>
<td>RMSE (N/m) RMSRE</td>
<td>RMSE (M/m) RMSRE</td>
</tr>
<tr>
<td>Raw</td>
<td>yes N/A</td>
<td>59 65%</td>
<td>174 53%</td>
</tr>
<tr>
<td>Simple Gravity</td>
<td>no N/A</td>
<td>85 62%</td>
<td>260 54%</td>
</tr>
<tr>
<td>Simple Gravity</td>
<td>yes const</td>
<td>27 22%</td>
<td>9 2%</td>
</tr>
<tr>
<td>Simple Gravity</td>
<td>yes linear</td>
<td>28 24%</td>
<td>4 1%</td>
</tr>
<tr>
<td>Simple Gravity</td>
<td>yes root</td>
<td>25 21%</td>
<td>4 1%</td>
</tr>
<tr>
<td>General Gravity</td>
<td>no N/A</td>
<td>40 31%</td>
<td>117 24%</td>
</tr>
<tr>
<td>General Gravity</td>
<td>yes const</td>
<td>16 13%</td>
<td>4 1%</td>
</tr>
<tr>
<td>General Gravity</td>
<td>yes linear</td>
<td>16 13%</td>
<td>4 1%</td>
</tr>
<tr>
<td>General Gravity</td>
<td>yes root</td>
<td>15 12%</td>
<td>4 1%</td>
</tr>
</tbody>
</table>
Summary: Tomo-gravity Works

- Tomo-gravity takes the best of both tomography and gravity modeling
  - Simple, and quick
  - A few seconds for whole AT&T backbone
  - Satisfies link constraints
  - Gravity model solutions don’t
  - Uses widely available SNMP data
    - Can work within the limitations of SNMP data
    - Only uses first order statistics $\rightarrow$ interpolation very effective
  - Limited scope for improvement
    - Incorporate additional constraints from other data sources: e.g., Netflow where available

- Operational experience very positive
  - In daily use for AT&T IP network engineering
  - Successfully prevented service disruption during simultaneous link failures

Summary: Traffic Matrix Estimation

- TM estimation is highly under-determined
- Estimation accuracy depends on
  - model assumptions
  - prior distributions
  - additional information
    - partial data, routing, topology
- New applications
  - Detect anomalies using traffic matrix time series
  - “Network Anomography” Yin Zhang, et al, IMC 2005