Linear Algebra for Network Loss Characterization

David Bindel

UC Berkeley, CS Division

Application: Network monitoring

- Set of n hosts in a large network
- $n(n-1)/2$ (undirected) paths between them
- Want latency and packet loss rates for each path
- Use info to choose servers, route around faults

Network monitoring related work

- Network distance estimation systems (RON, GNP)
- Network tomography systems
- Latency inference work (Shavitt et.al.)

Loss rates as distance

- Assume link losses are independent
- $P(\text{packet traverses path}) = \prod P(\text{packet traverses links})$
- $-\log P(\text{packet traverses path}) = \sum -\log P(\text{packet traverses links})$
- $-\log P(\text{transmission success})$ is a distance measure

Path representation

Network has s links. Represent paths by vectors $v \in \mathbb{R}^s$:

$$v_i = \begin{cases} 1 \text{ if link } i \text{ is used on the path} \\ 0 \text{ otherwise} \end{cases}$$

Let $x_i = -\log P(\text{transmission success on link } i)$. Then

$$-\log P(\text{packet loss on path}) = v^T x_i$$

Path representation

Care about $r = n(n-1)/2$ paths. Let the rows of $G \in \mathbb{R}^{r \times s}$ represent paths, and $b \in \mathbb{R}^r$ represent path losses

$$
\begin{aligned}
G_{ij} &= \begin{cases} 1 \text{ if link } j \text{ is used on the path } i \\ 0 \text{ otherwise} \end{cases} \\
b_i &= -\log P(\text{transmission success on path } i)
\end{aligned}
$$

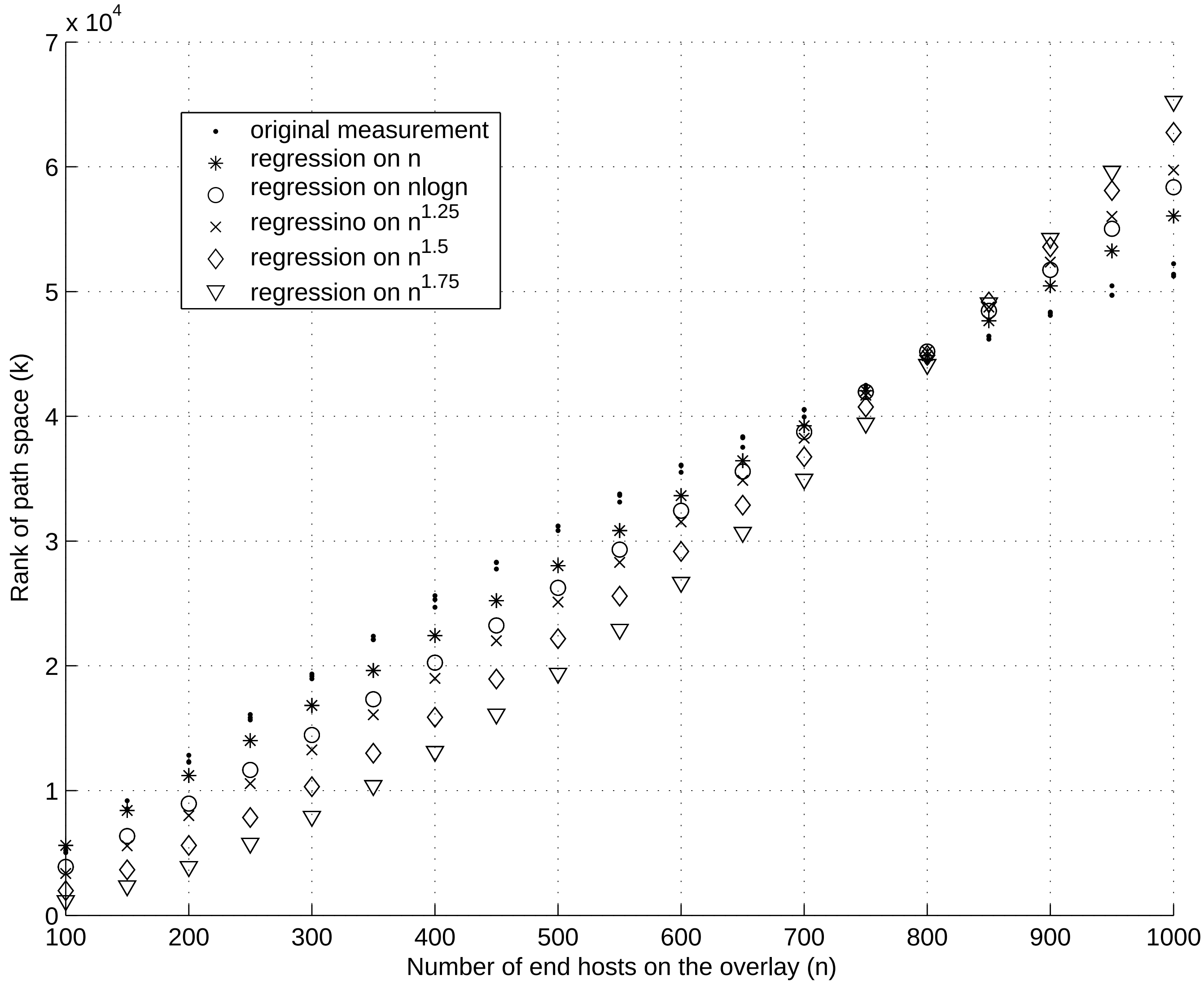
Then path losses are related to link losses by

$$Gx = b$$

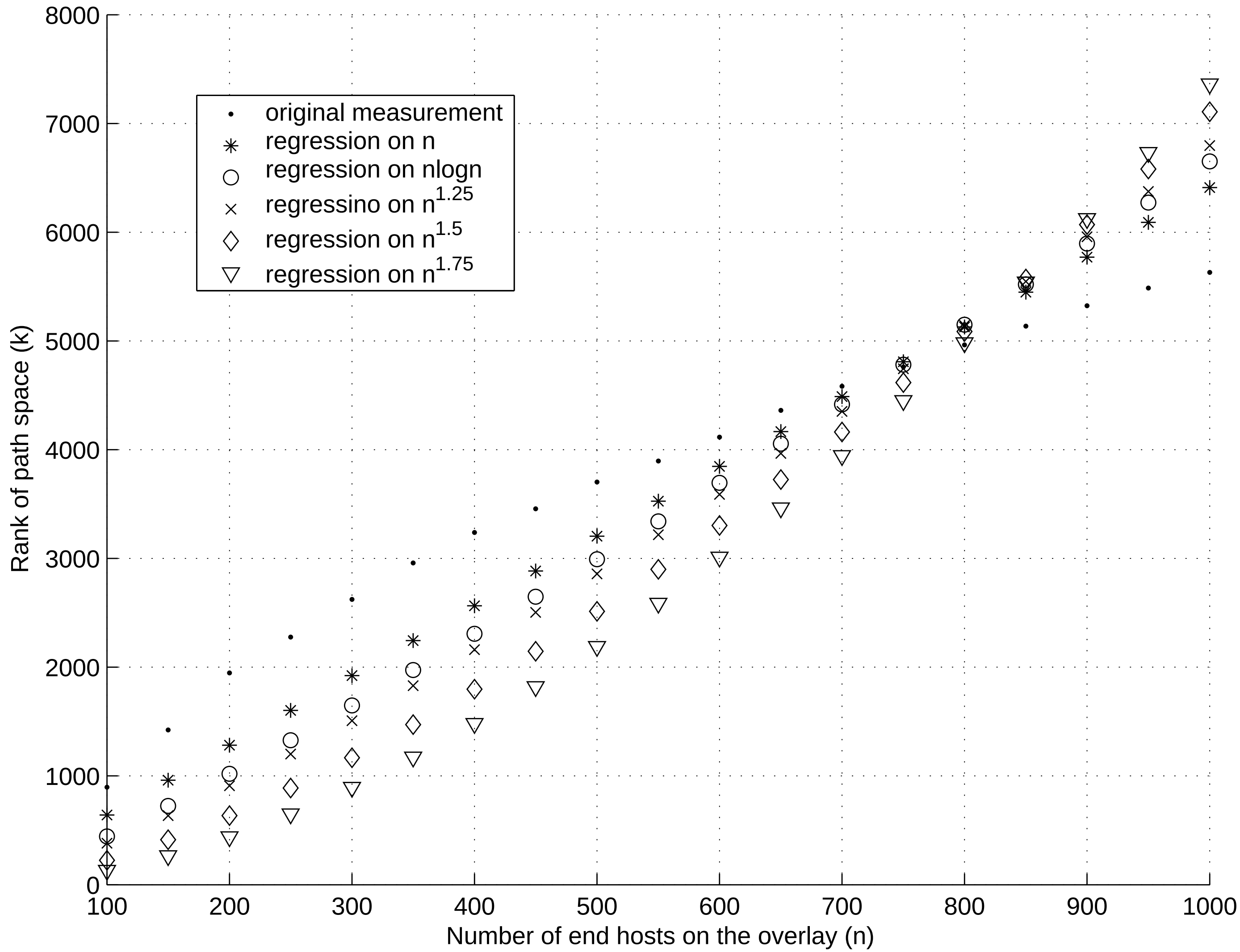
Rank of G

- The Internet has moderately hierarchical structure
- Paths overlap in many links
- $k := \text{rank}(G) \leq$ links used $=$ nonzero columns of G
- links used $< O(n^2)$ – seems to grow like $O(n)$ or $O(n \log n)$
- k is usually less than number of links used

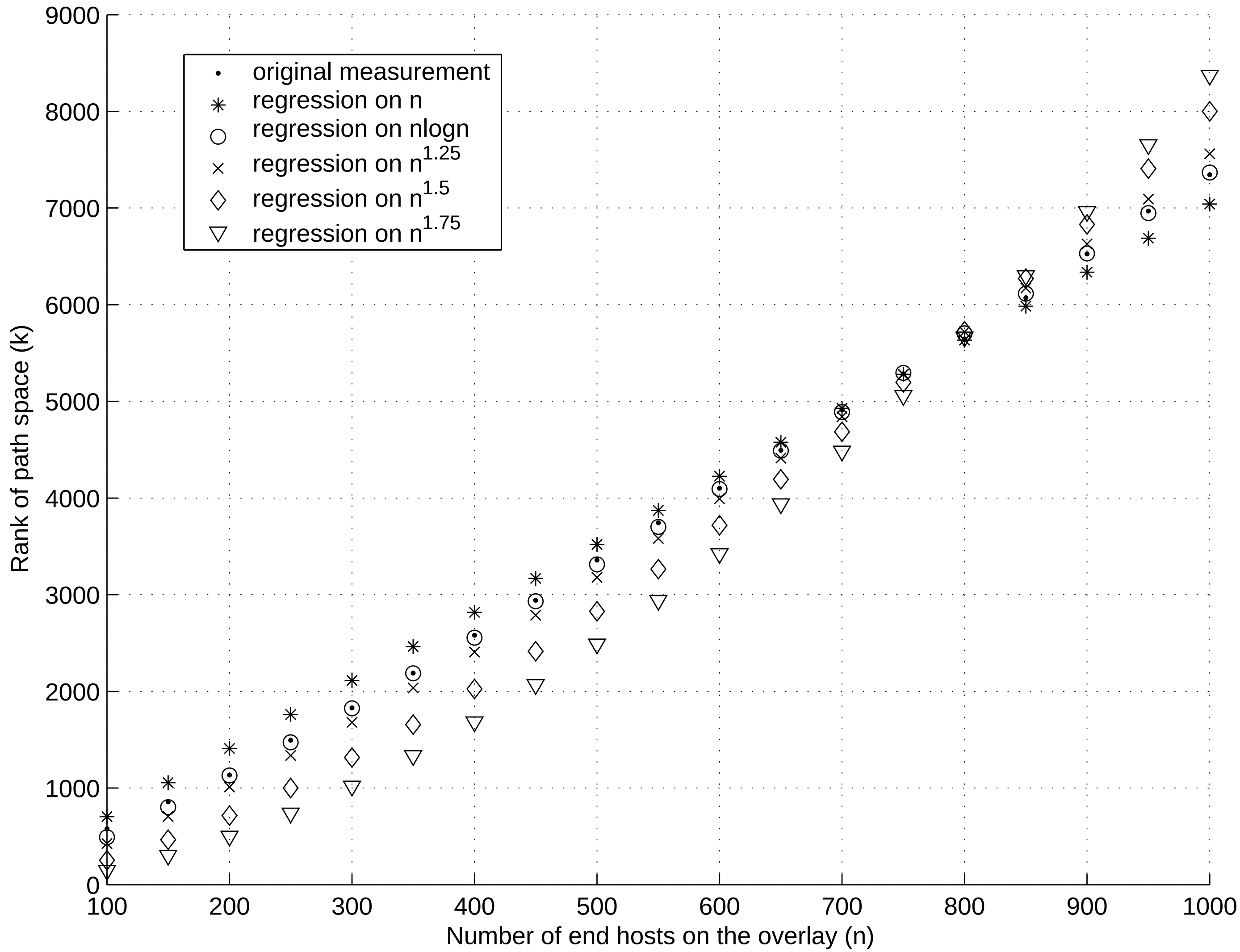
Rank of G: Lucent scan (bound)



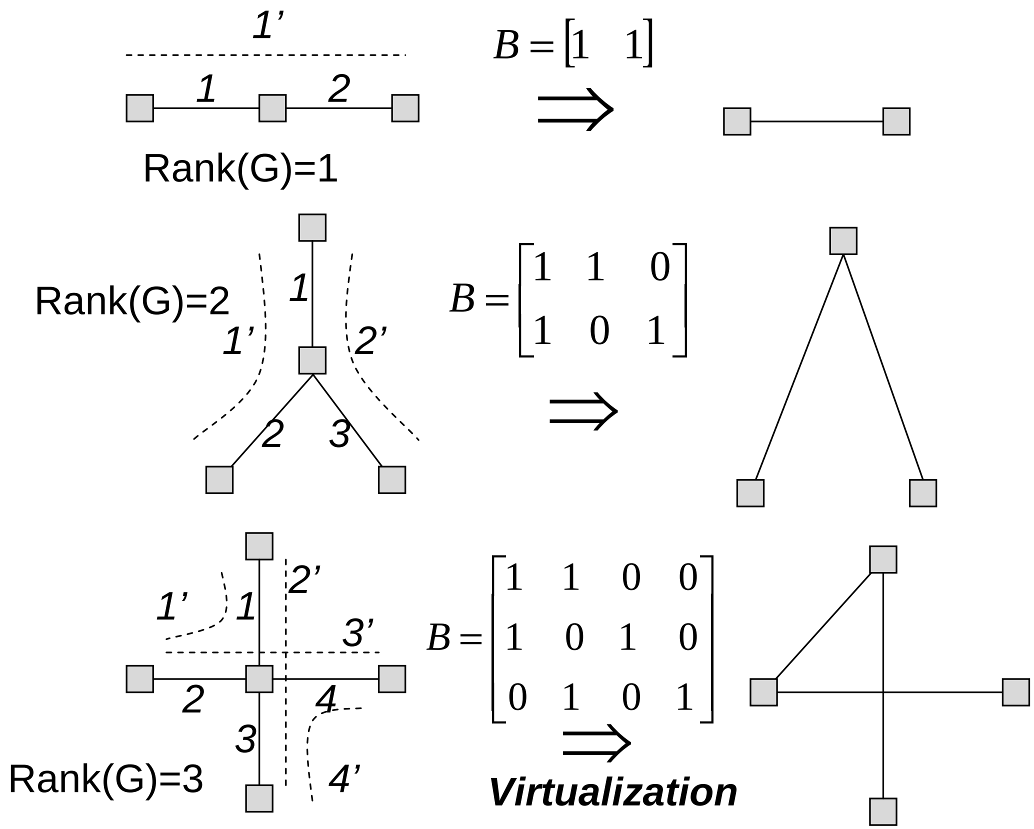
Rank of G: AS-level Albert-Barabasi



Rank of G: AS Barabasi + RT Waxman



Virtualization and local elimination



Real links (solid) and overlay paths (dotted) going through them

Virtual links

Path loss inference

- Rank deficiency in G implies solutions to $Gx = b$ are non-unique
- Choose k independent rows of G ($\bar{G}$) and of b ($\bar{b}$)
- Monitor k paths to estimate $\bar{b}$
- Compute any solution to $\bar{G}x = \bar{b}$
- Compute $b = Gx$ for rest of loss rates

Indentifiable paths

- Rows of $\bar{G}$ form a basis for the row space of G
- Path vectors in the row space of $\bar{G}$ are *identifiable*:
  - All vectors for end-to-end paths are identifiable
  - Sums of identifiable paths are identifiable
  - Some links are identifiable; many are not
- Identifiable path loss rates can be inferred from $\bar{b}$

Bounding unidentifiable path losses

Let v represent an unidentifiable path which transmits packets with probability p. Get bounds on p using fact that $b \geq 0$ and $x \geq 0$:

1. If $w = \bar{G}^T c \geq v$ then $w^T x = c^T b \geq v^T x = -\log(p)$
2. If $w = \bar{G}^T c \leq v$ then $w^T x = c^T b \leq v^T x = -\log(p)$

So bound $\log(p)$ by solving two linear programming problems:

1. Minimize $c_u^T b$ subject to $G^T c_u \geq v$
2. Maximize $c_l^T b$ subject to $G^T c_l \leq v$

Properties of G

- Very sparse
- R factor in QR mostly full (tried some reordering)
- Observed $\kappa = O(100)$ (discarding singular directions)

Iterative methods should work well; have not yet coded.

Algorithm tasks

- Choose basis $\bar{G}$ for row space of G
- Solve linear systems involving $\bar{G}$
- Quickly update basis choice / factorizations on:
  - Addition of new nodes / paths
  - Deletion of nodes / paths
  - Localized changes to network topology

Key ingredient is quickly solving linear systems with $\bar{G}$.

Current algorithm

- QR factorization of G^T
- Only keep part of R for $\bar{G}^T$
- Basically block CGS with iterativerefinement
- Store R densely, but in (block) packed single precision
- Use $Q := G^T R^{-1}$ if needed

Row selection and factorization

Input: Current $\bar{G}$, path vectors V, current R
Output: Updated $\bar{G}$, R

```
R12     = R' \ (Gbar' * V);
R22     = V' * V - R12' * R12;
[q,r,e] = qr(R22);
k       = sum(abs(diag(r)) > tol);
R       = [R, R22(:,       e(1:k));
           0, R12(e(1:k), e(1:k));
Gbar    = [Gbar; V(e(1:k),:)'];
```

Computing x

- Choose minimum norm solution to $\bar{G}x = \bar{b}$
- $x = (\bar{G}^T R^{-1}) R^{-T} \bar{b}$ plus iterative refinement
- Path v identifiable if $\|R^{-T} \bar{G} v\| = \|v\|$

Removing paths

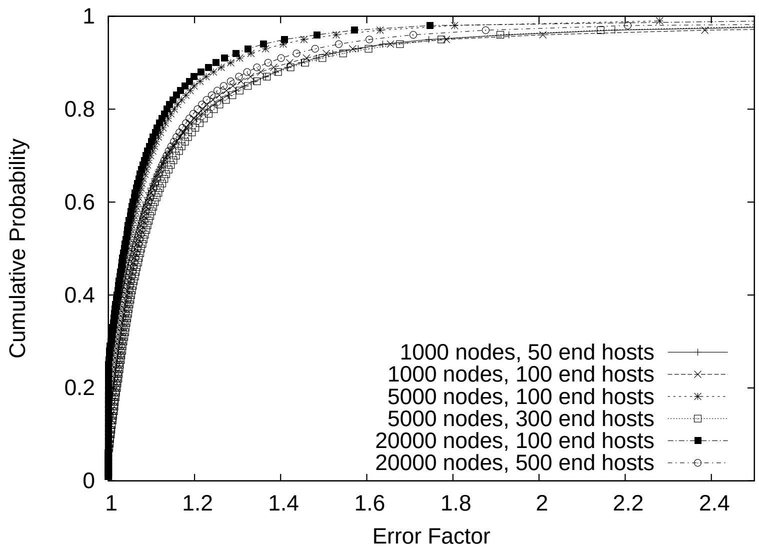
Paths not in $\bar{G}$ are trivial. Paths in $\bar{G}$ are trickier. To remove row i of $\bar{G}$:

- Compute a vector in the null space of $\bar{G}$ minus row i: $\bar{G}_{\text{orig}} y = e_i$
- Compute $r := Gy$
- If $r \neq 0$ then add row j such that $r_j \neq 0$ to $\bar{G}$
- If $r = 0$ then k decreased by one, no replacement
- Can update R in $O(k^2)$ time in standard way

Some system issues

- Measurement load balancing
- Construction and updates to G are nontrivial
  - Incorrect mapping due to aliasing is okay.
  - Can still do useful work with incomplete info.
- How do applications actually use the info?
  - Set up notifications when a path becomes lossy
  - Query server for loss rates when choosing paths

Simulation results



- Loss rates based on 10K samples / path
- Bernoulli model (independent trials) or Gilbert model (correlated trials – bursty)
- Plot (relative) error in p vs. measurement
- Haven't analyzed expected error yet

Conclusions and future work

- Experiments with PlanetLab testbed in progress
- Make code available to other researchers
- Further explore combinatorial structure of the problem
- Test out iterative methods
- Actually implement linear program based bounds
- Distribute work among servers