Complex-Network Modelling and Inference

Lecture 25: Network Topology Inference

Matthew Roughan
<matthew.roughan@adelaide.edu.au>
https://roughan.info/notes/Network_Modelling/

School of Mathematical Sciences,
University of Adelaide

January 14, 2025

Section 1

Stochastic Tomography on General Networks

The general inference problem

- We have an arbitrary, but known network
  - we also know routes used on the network
  - we express this through a *routing matrix* R
    often it is denoted by A but I want to keep A for adjacency matrices
- We have some end-to-end measurements
  - *e.g.*, delay of packets
  - *e.g.*, number of successful packets
- We have a statistical model
  - *e.g.*, packet loss is an independent Bernoulli process at each link
  - *e.g.*, packet delays are independent Gaussian RVs at each link

These models are approximations!!!

An Aside

- Packet losses and (large) delays have varying causes, but the typical assumption is that they are caused by *congestion*
- Each router has multiple input and output links
  - typically, we might have the sum of the input equal to the sum of the output capacities
  - but at any one time, the flow of traffic may funnel towards a single (or small set) of outputs
  - there are queues (buffers) at each input (or output or both) to absorb this for a transient overload
- If the overload continues for some time (say a few seconds), the buffers fill
  - packets are delayed (significantly more than standard transmission delays)
  - if the buffers fill completely, packets overflow, and are dropped
- Some loss/delay happens all the time, at random
  - *e.g.*, a few tens of ms of propagation delay
  - *e.g.*, < 1 % of packets are dropped

but sometimes things get bad

Notation: Delay

- x_i is the average delay at link i
  - it doesn't have to be the average, but let's keep it simple
- y_j is the average delay along end-to-end path j
- R is the *routing matrix*, which is (usually) a 0-1 *incidence matrix* where

$$R_{ij} = \begin{cases} 1 & \text{if link } j \text{ appears on path } i, \\ 0 & \text{otherwise.} \end{cases}$$

- We assume delays add up along a path

Notation: Packet Loss

- x_i is the negative-log transmission (success) probability at link i
- y_j is the negative-log transmission (success) probability along end-to-end path j
- R is the *routing matrix*, which is (usually) a 0-1 *incidence matrix* where

$$R_{ij} = \begin{cases} 1 & \text{if link } j \text{ appears on path } i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the probability of successful transmission of a packet over independent links is the product of the individual probabilities, so

$$P\{\text{success on link } i_1 \text{ and } i_2\} = P\{\text{success on link } i_1\} \times P\{\text{success on link } i_2\},$$

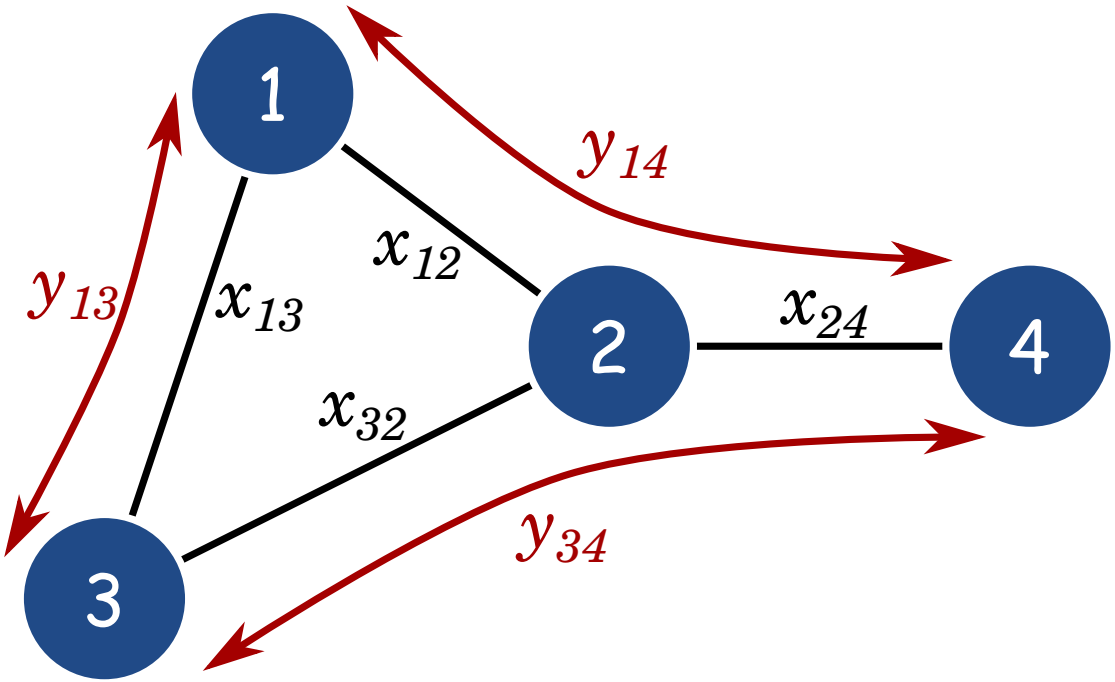
and so

$$-\log P\{\text{success on link } i_1 \text{ and } i_2\} = x_{i_1} + x_{i_2}$$

Once again, these are additive

Simplifications

- won't worry about direction (assume its symmetric)
- shortest-path (minimum hop) routing
- I will assume I can only use nodes 1,3 and 4 as end-points (otherwise the problem is trivial)



$$y_{13} = x_{13}$$

$$y_{14} = x_{12} + x_{24}$$

$$y_{34} = x_{32} + x_{24}$$

Express the Tomography Equations in Matrix Form

Replace

$$\begin{aligned} y_{13} &= x_{13} \\ y_{14} &= x_{12} + x_{24} \\ y_{34} &= x_{32} + x_{24} \end{aligned}$$

with

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

where R is the routing matrix

$$R = \begin{array}{c} \\ y_{13} \\ y_{14} \\ y_{34} \end{array} \begin{array}{c} \begin{array}{cccc} x_{12} & x_{13} & x_{32} & x_{24} \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

$$y = Rx$$

- Looks easy to solve, but
  - note there are more unknowns than measurements
  - even when R is square, it is usually not full rank
  - there is measurement noise, so really

$$y = Rx + \varepsilon$$

- There are many methods to solve this
  - they need to take into account that the equations are
    - ★ under-constrained
    - ★ inconsistent
  - approach: use a model
    - ★ Bayesian prior
    - ★ Max-entropy prior
    - ★ Max-sparsity prior

    I think of them all as *regularisation* techniques

Solution by regularisation

Problem:

$$\mathrm{y} = R\mathrm{x} + \varepsilon$$

where

- we know y
- want to estimate x

Solution: pose it as an optimisation problem

$$\hat{\mathrm{x}} = \underset{\mathrm{x}}{\operatorname{argmin}} \|R\mathrm{x} - \mathrm{y}\|_2^2 + \lambda d(\mathrm{x}, MODEL),$$

where $d(\cdot, \cdot)$ is a distance measure, and λ a parameter we have to choose:

- L_2 norm $\|R\mathrm{x} - \mathrm{y}\|_2^2$ linked to "belief" that noise ε is Gaussian
- large λ assumes noise ε is large, and/or model is good
- small λ puts more faith in the data
- specific form of $d(\cdot, \cdot)$ is problem specific, but ideally should be convex, so that the optimisation is practical

Solution by regularisation

- We can think of solutions as being in this form, but we still need
  - the model
  - a method to solve the optimisation
    - ★ *e.g.*, In MLE (Max Likelihood Estimation) shown below we are maximising a likelihood "distance" but we need a practical algorithm to do so
- There is always an underlying ambiguity in the data
  - we can only get a statistical solution
  - it can only be as good as the data + the model

Section 2

(Stochastic) Loss Estimation on Trees

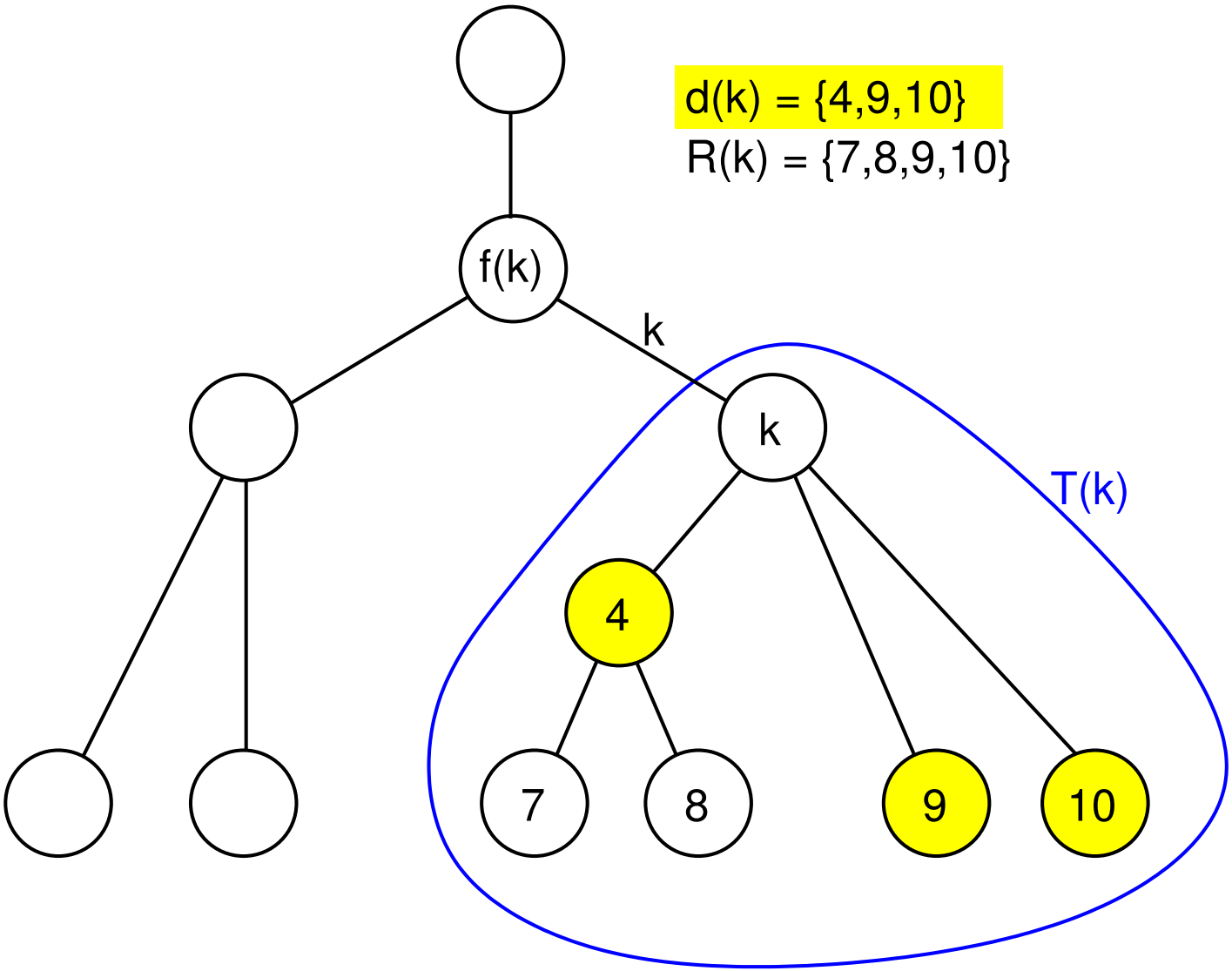
Example MLE for loss measurement

- Topology again reduced to a tree
- Implicit model is that loss happens as high in the tree as possible
- Notation

$$
\begin{aligned}
A_k &= P\{ \text{ packet reaches node } k \ \} \\
T(k) &= \text{ the dependent tree rooted at node } k \\
f(k) &= \text{ the parent of node } k \\
&= \text{ link } (f(k), k) \text{ is just called link } k \\
\alpha_k &= P\{ \text{ packet crosses link } k \ \} = A_k / A_{f(k)} \\
d(k) &= \text{ children of node } k \\
R(k) &= \text{ the leaves of the tree, } \textit{i.e.}\text{, receivers, under node } k \\
\gamma_k &= P\{ \text{ packet reaches at least one node of } R(k) \}
\end{aligned}
$$

- If a packet is loss at link k, it won't be seen anywhere below

Example MLE for loss measurement



Example MLE for loss measurement

Measurements:

$$\hat{X}_m^{(k)} = \begin{cases} 1 & \text{if the } m\text{th packet is received at any node } \in R(k) \\ 0 & \text{otherwise} \end{cases}$$

which can be formed recursively by

$$\hat{X}_m^{(k)} = \begin{cases} X_m^{(k)} & \text{if } k \in R \\ \max_{j \in d(k)} X_m^{(j)} & \text{otherwise} \end{cases}$$

where

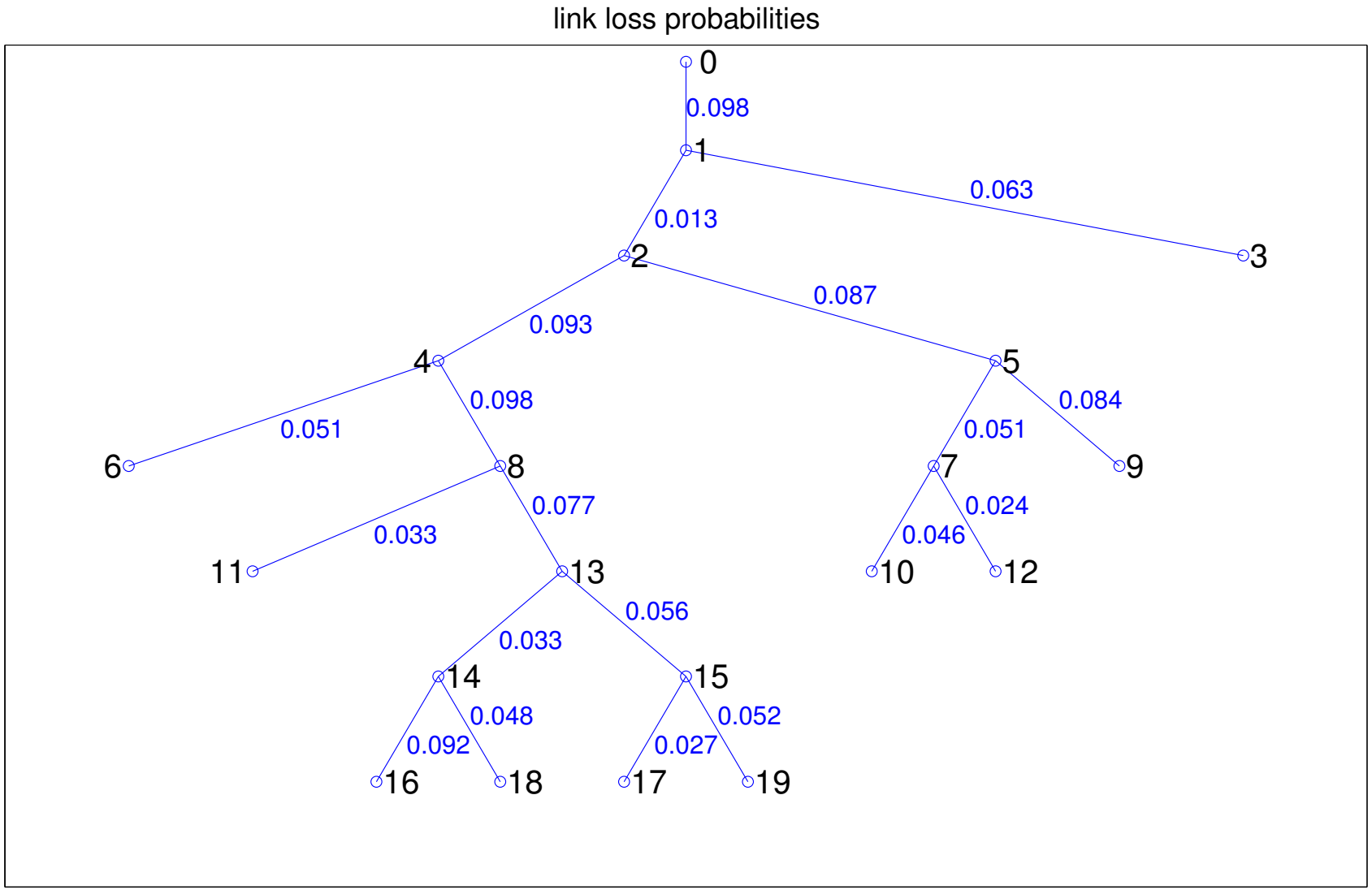
$$X_m^{(k)} = \begin{cases} 1 & \text{if the } m\text{th packet is received at node } k \\ 0 & \text{otherwise} \end{cases}$$

Now, treat the $\hat{X}_m^{(k)}$ as a set of measurements of a Bernoulli RV, and use the MLE for the probability, and we get

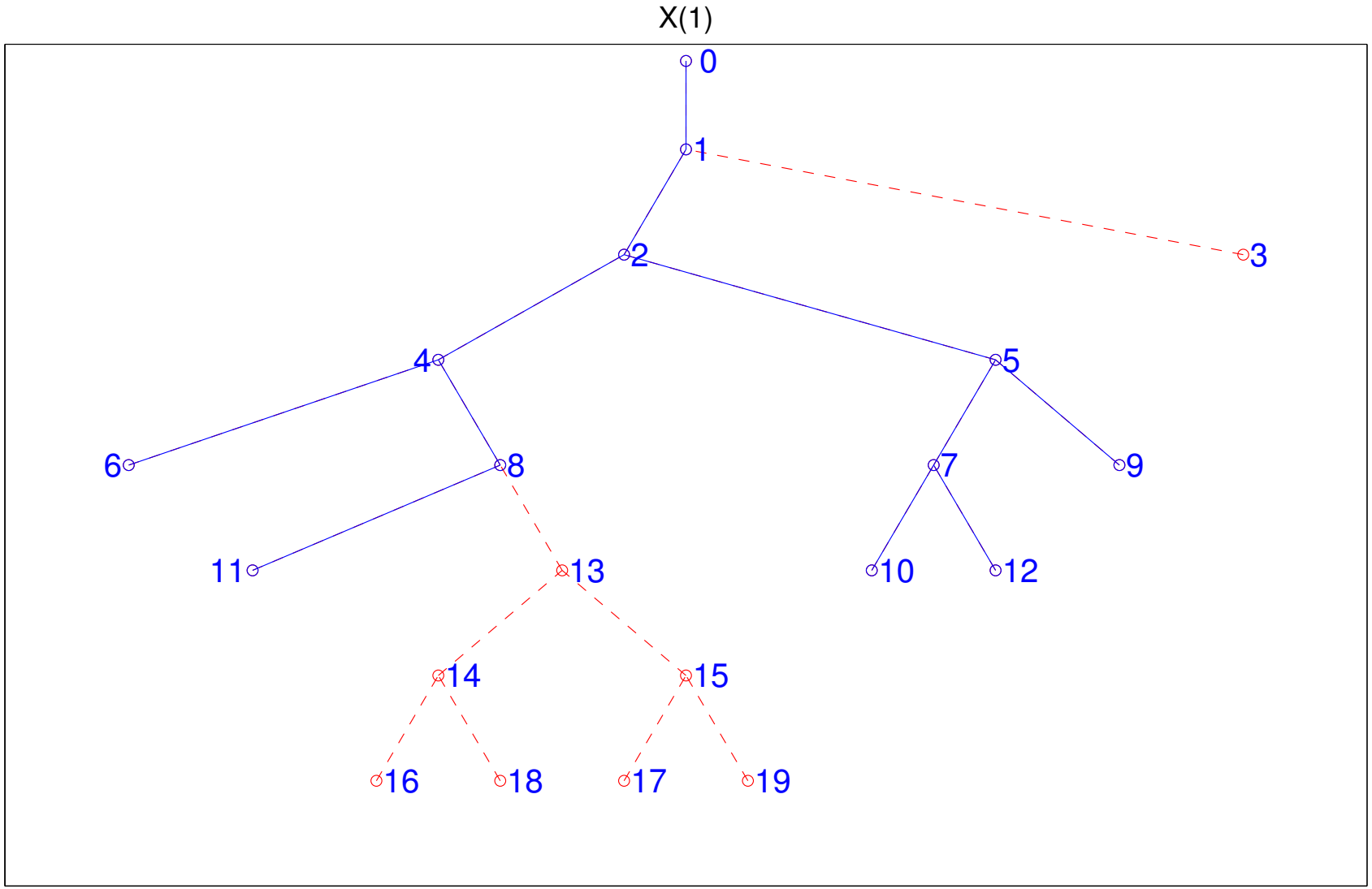
$$\hat{\gamma}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{X}_m^{(k)}$$

which we evaluate recursively.

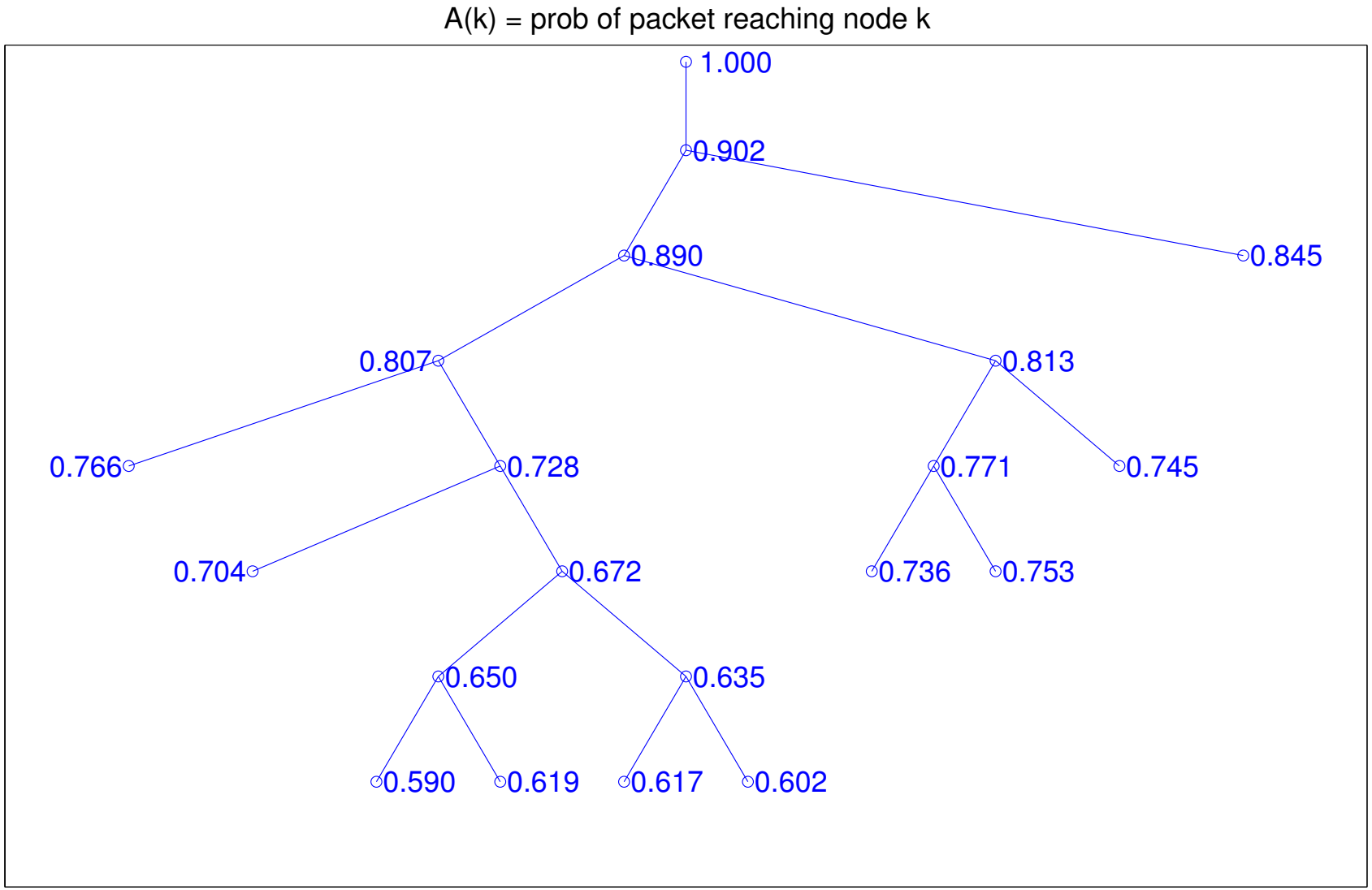
Example (10,000 simulated probes)



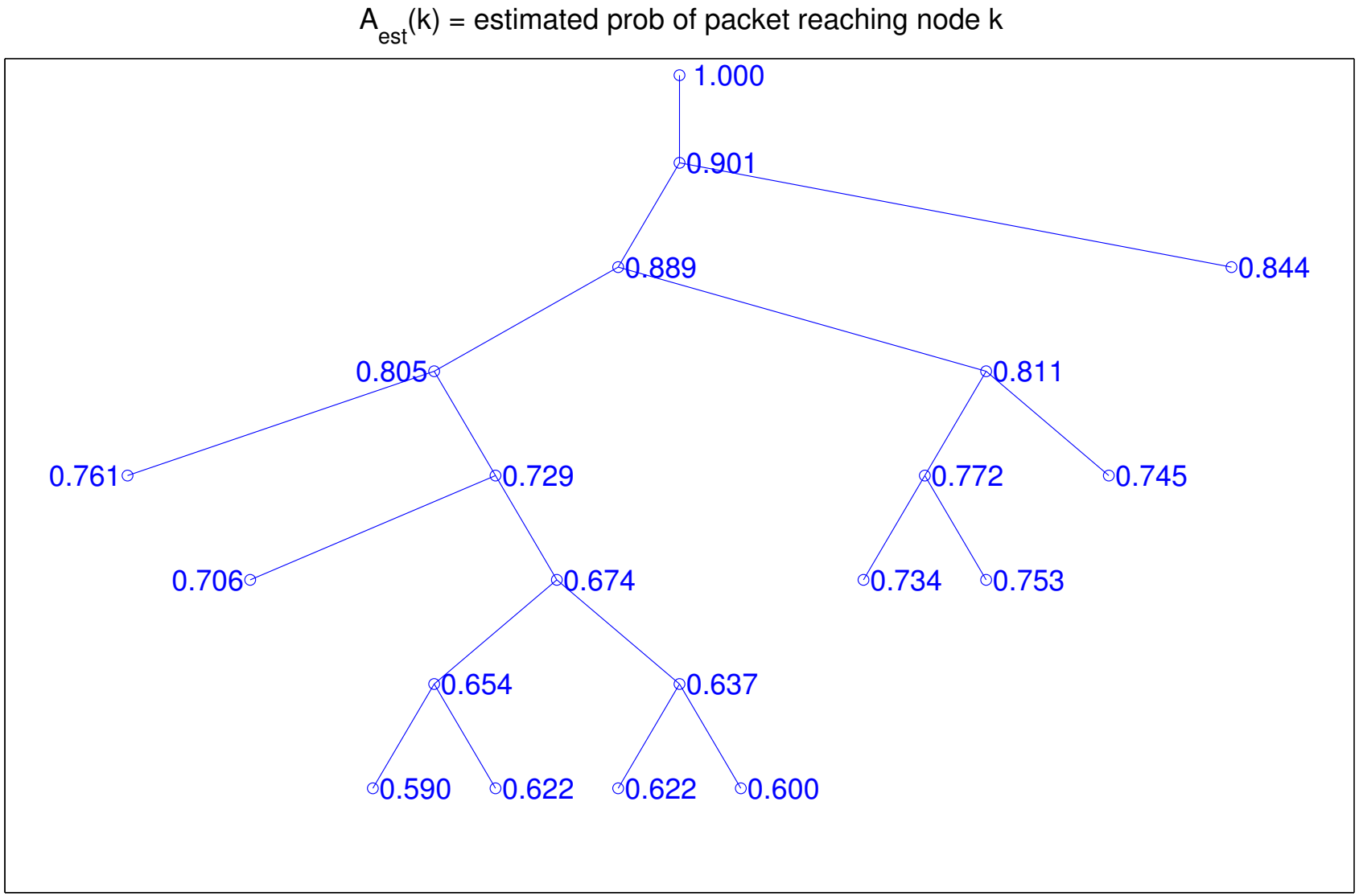
Example (10,000 simulated probes)



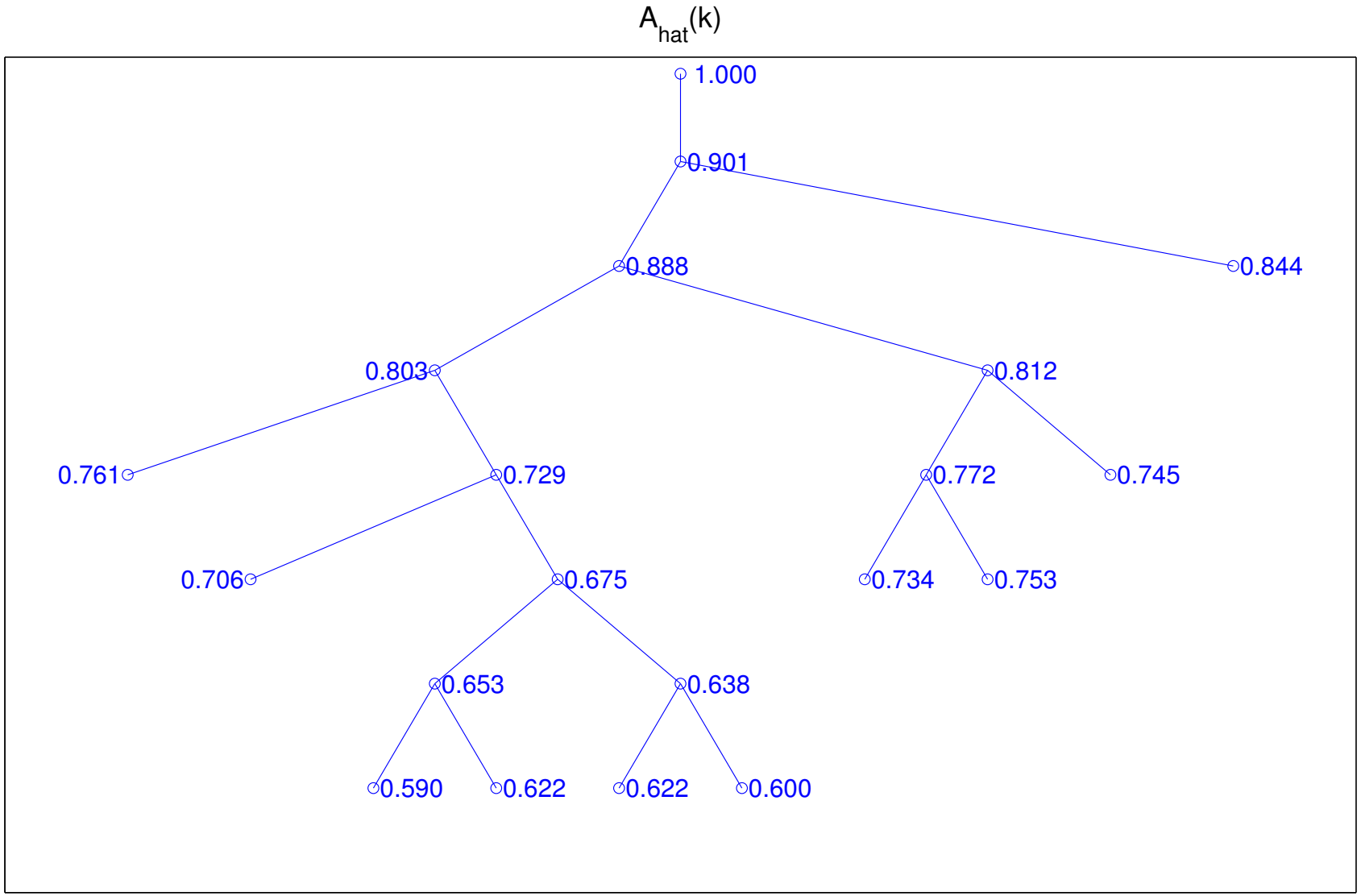
Example (10,000 simulated probes)



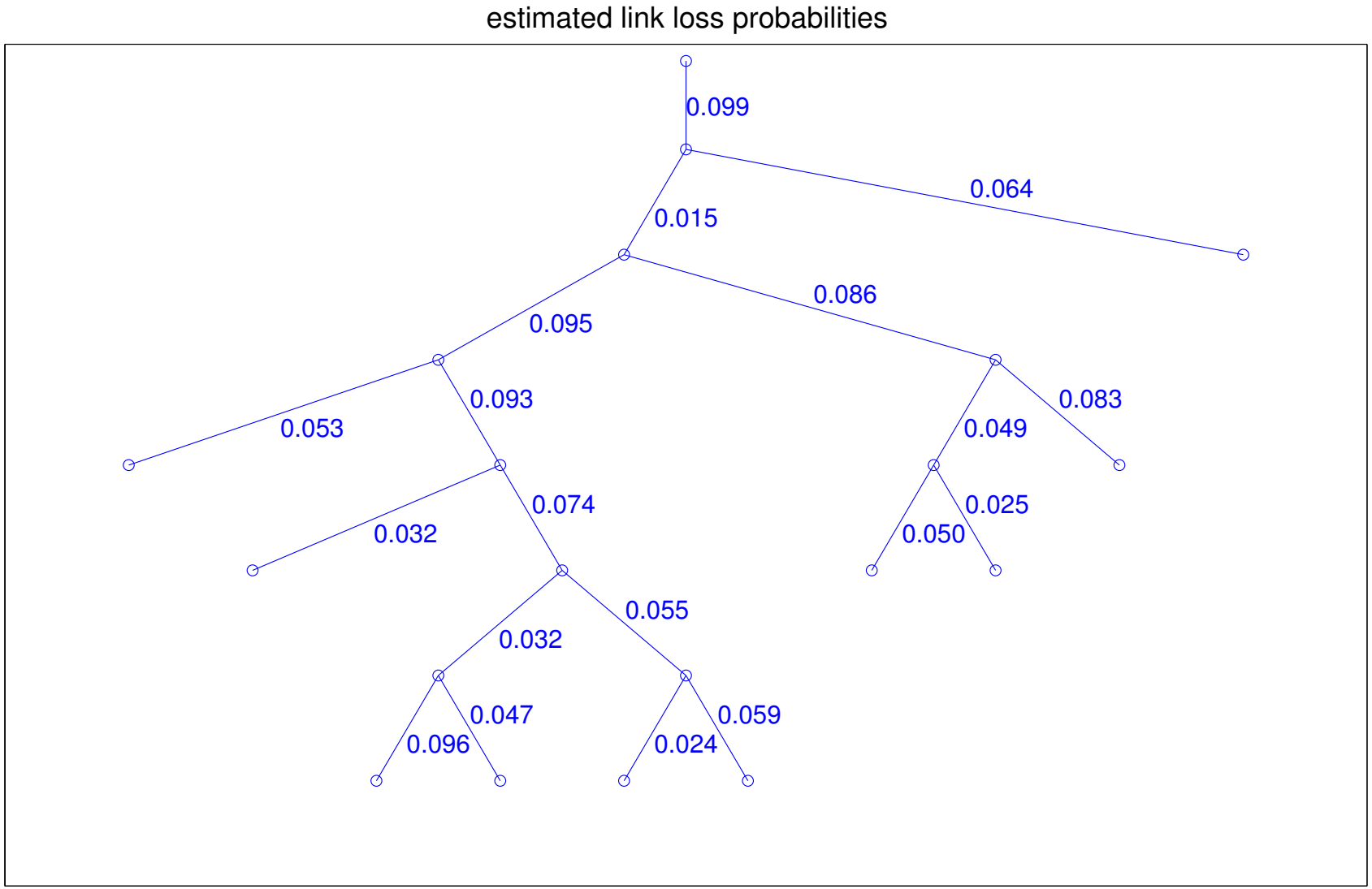
Example (10,000 simulated probes)



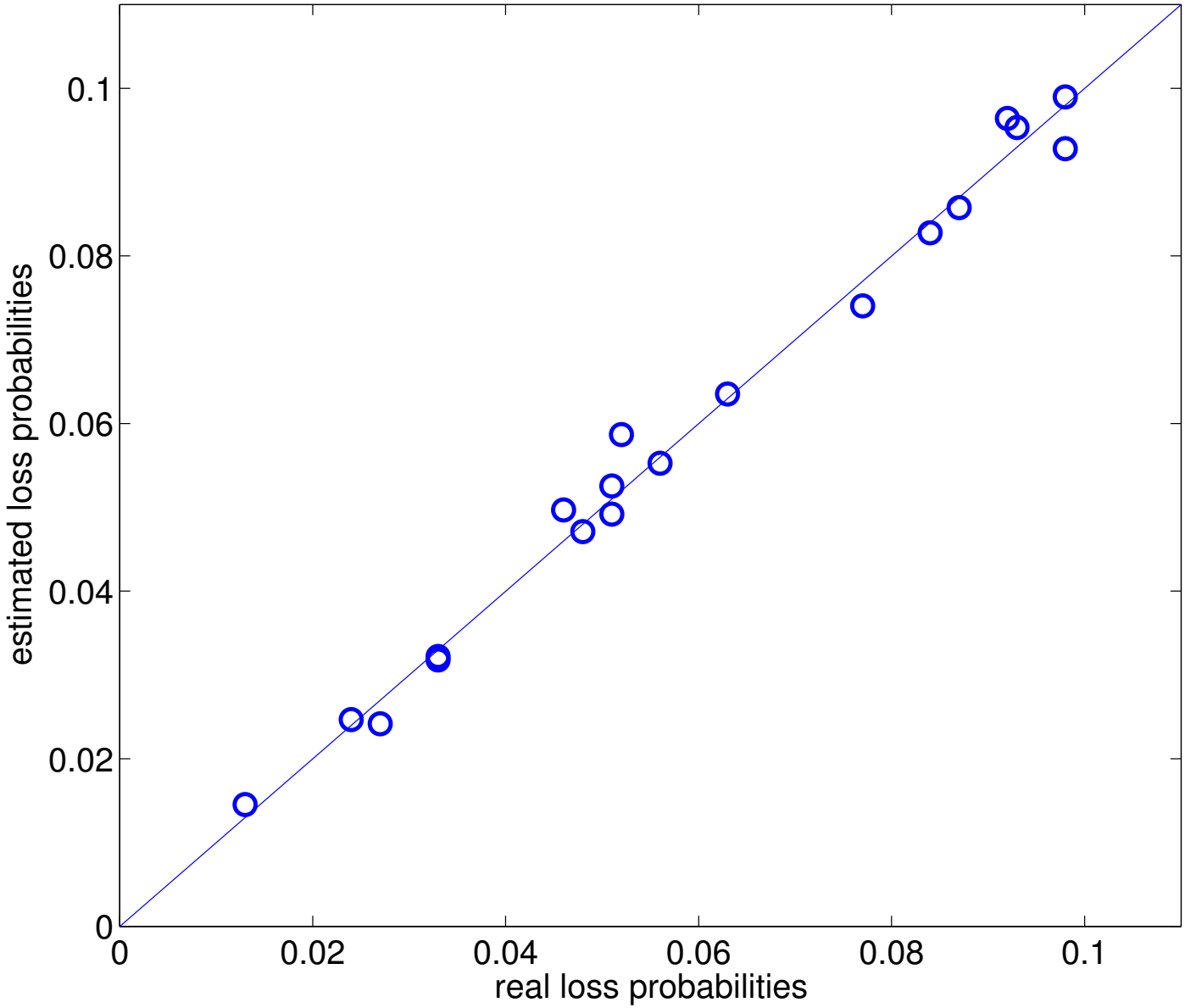
Example (10,000 simulated probes)



Example (10,000 simulated probes)



Example: summary of results



Section 3

Inferring the Network

Inferring Trees

- Our approach is a MLE, *i.e.*, we find

$$\hat{\alpha}_{\mathcal{T}} = \underset{\alpha}{\text{argmax}}\, \mathcal{L}(\mathcal{T}, \alpha)$$

where $\mathcal{L}(\mathcal{T}, \alpha)$ is the log-likelihood of the loss tree (given the measurements $X_m^{(k)}$)

$$\mathcal{L}(\mathcal{T}, \alpha) = \log P\left\{ X_m^{(k)},\ m = 1, \ldots, n,\ k = 1, \ldots, |V| \middle| \mathcal{T}, \alpha \right\}$$

where there are n measurements, and $|V|$ nodes, and the tree $\mathcal{T}$ is known, and we assume independent Bernoulli losses.

- The work above is to compute it efficiently, using a recursive structure, and pushing loss up through the tree.

Inferring Trees

To infer the best tree, just use the consistent MLE

$$\hat{\mathcal{T}}_{ML} = \underset{\mathcal{T} \in \mathcal{T}(\mathcal{R})}{\operatorname{argmax}} \mathcal{L}(\mathcal{T}, \hat{\alpha}_{\mathcal{T}})$$

That is, find the tree that maximises the likelihoods.

We might imagine taking each possible tree, solve it as above, and then choose the one whose resulting ML is the best.

Inferring Trees

Given 1 source, n leaves, and m intermediate nodes, how many trees are possible?

Simpler, how many trees with n (labelled) nodes?

Inferring Trees

Given 1 source, n leaves, and m intermediate nodes, how many trees are possible?

Simpler, how many trees with n (labelled) nodes?
Cayley's formula n^{n-2}

Inferring Trees

Even if we limited ourselves to possible trees, there are too many to test for even a moderate sized problem.
So we need to be clever

Heuristic approaches

- Significant loss often occurs at a few bottlenecks
- Receivers with similar loss probabilities might be because they have the same bottleneck
  - actually want to be a bit cleverer
  - cluster based on a transform of the loss rates
- So form the tree by hierarchical clustering
  - multiple algorithms for clustering

Direct Statistics Approaches

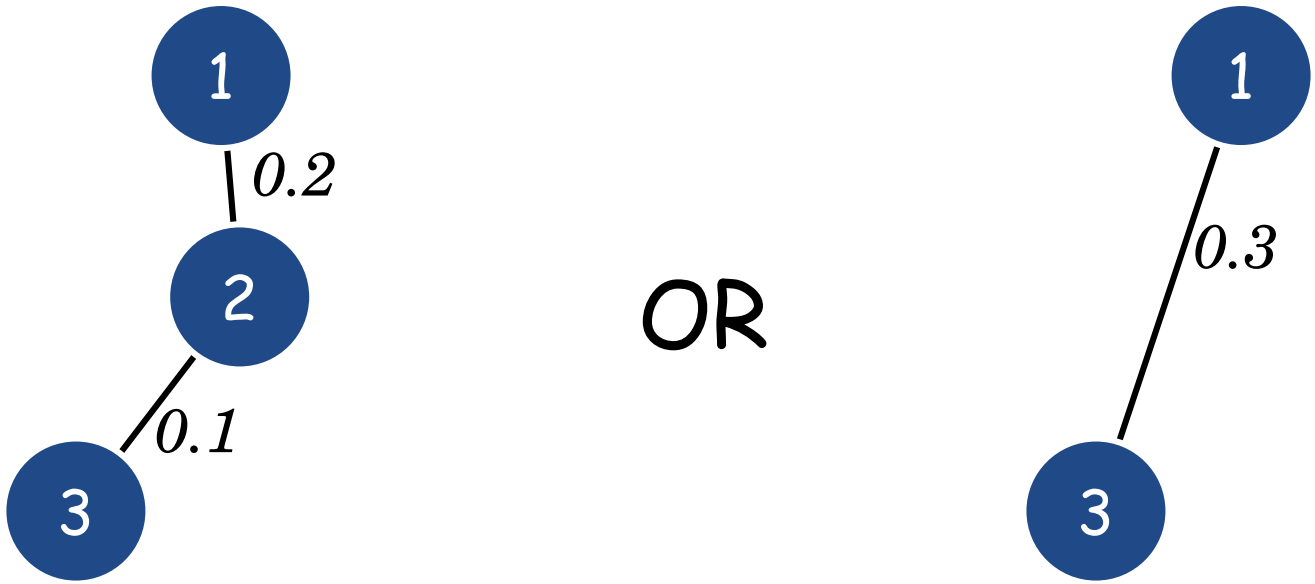
- EM (Expectation Maximisation)
- Gaussian Mixture Models

Both have been tried.

Intractable Issues

Identifiability can be problematic

- *e.g.*, if you can only measure the path $1-3$, you can never tell the difference between these two (delay) networks



Conclusion

- Once we have an estimator (at least an MLE) for performance, we can extend this to estimate the MLE tree
- There is a challenge in terms of computation
- Techniques have been applied to trees
  - general networks are much harder
  - there are big research problems here

Further reading I