Complex-Network Modelling and Inference
Lecture 13: Random Graphs: HOT and COLD

Matthew Roughan
<matthew.roughan@adelaide.edu.au>
Network_Modelling/

School of Mathematical Sciences,
University of Adelaide

October 18, 2017
Section 1

HOT graphs
Problem with “random” graph

- Random graphs are formed by taking lots of small, random operations, and building a larger graph
  - most dependencies are quite local
  - bottom-up construction
  - hope for “emergent” behaviour
  - lots of small, local behaviour produces global structure

- This is NOT how real, technological, physical networks arise
  - they are designed
  - often from the top down
  - often optimised against a set of constraints

- Even if a random network has all the metrics right, does that mean it is actually a good model?
  - for instance, most random graph models don’t include any redundancy component
  - consequently, they can be vulnerable to failures
  - real networks have designed redundancy
Example: Internet

- It was noticed early on that the “Internet” has a power-law degree [FFF99]
  - this was one of the motivators for preferential attachment
  - ignore some of the holes in the paper for the moment
- The preferential attachment model has “central” high-degree node
  - if one of these fails, a preferential-attachment network might become partitioned
  - is this a worry for the REAL Internet
HOT

\[ \text{HOT} = \text{Highly Optimised Tolerances} \]
\[ = \text{Highly Organised Tradeoffs} \]
\[ = \text{Highly Optimised Topology} \]
\[ = \text{Heuristically Optimised Topology} \]

- It's a generic theory related to emergence or power laws.
- The idea is that power-laws emerge from systems that have been highly optimised.
- This is a pretty superficial take on it – there is a lot more, but we only need to see how it applies to networks.
HOT graphs

- Assume that networks are design through an optimisation process
- For the Internet
  - nodes are *routers* and edges are links between them
    - objective is to minimise the cost of these
  - routers have a maximum number of *ports*
    - effectively a maximum node degree
    - *backbone* routers, have a few, high-speed ports, but are very fast
    - *edge* routers have many low-speed ports
  - capacity constraint
    - must have enough capacity to carry given traffic
Is HOT a random graph model?

- HOT is different from other random graph models
  - the operations aren’t random (mostly)
  - the optimisation is nearly deterministic
- The randomness comes from the environment
  - in this case the (random) traffic that must be carries
- Random traffic matrix

\[ T_{ij} = \alpha B_i B_j, \]

where the \( B_i \) are non-negative random variables.
It's hard to model costs, so instead flip the problem around, so instead:

\[ x_{ij} \propto B_i B_j \]

\[
\max_{\alpha} \sum_{i,j} \alpha x_{ij} = \max \sum_{i,j} \alpha B_i B_j \\
\text{s.t.} \sum_{i,j \in \mathcal{R}_i} x_{ij} \leq B_k, \forall k
\]
Figure 6: Five networks having the same node degree distribution. (a) Common node degree distribution (degree versus rank on log-log scale); (b) Network resulting from preferential attachment; (c) Network resulting from the GRG method; (d) Heuristically optimal topology; (e) Abilene-inspired topology; (f) Sub-optimally designed topology.
Abilene ecosystem of networks [LAWD04]

Figure 4: CENIC and Abilene networks. (Left): CENIC backbone. The CENIC backbone is comprised of two backbone networks in parallel—a high performance (HPR) network supporting the University of California system and other universities, and the digital California (DC) network supporting K-12 educational initiatives and local governments. Connectivity within each POP is provided by Layer-2 technologies, and connectivity to the network edge is not shown. (Right): Abilene network. Each node represents a router, and each link represents a physical connection between Abilene and another network. End user networks are represented in white, while peer networks (other backbones and exchange points) are represented in gray. Each router has only a few high bandwidth connections, however each physical connection can support many virtual connections that give the appearance of greater connectivity to higher levels of the Internet protocol stack. ESnet and GEANT are other backbone networks.
5 DISCUSSION

In the core of the problem lies the notion that high-performance networks are also likely to be robust. However, we have shown that it is quite possible to identify probabilistically rare and poorly performing networks. This is a significant finding because it challenges the prevailing wisdom that high-performance networks are necessarily robust.

Figure 6 shows that graphs having the same node degree distribution can be very different in their structure, particularly when viewed under the lens provided by the majority of the current performance data. For example, the graphs corresponding to the so-called "scale-free" models in Figure 8 are essentially the same in terms of their performance. However, there is a striking contrast observed by the literature [4, 42], which we believe is due to the story related to performance and likelihood.

We also notice that low likelihood itself does not guarantee a high performance network, as the network in Figure 6(f) shows that it is possible to identify probabilistically rare and poorly performing networks. However, based on current evidence, it does appear to be more difficult to find a relatively good design using random rewiring. Despite the fact that all of these graphs have the same overall degree distribution, the performance varies significantly.

The high degree "hubs" saturate in the PA and GRG networks, all routers achieve high utilization in the HOT network, whereas, when however the PA and GRG models cannot. Figure 7(d) shows that the HOT network can support users with a wide range of bandwidth requirements, its best performance. Figure 7 (a) shows that the HOT network is the best performer among the other networks having the same node degree distribution.

The PA and GRG networks have high likelihood, whereas the HOT network has high performance. The sub-optimal network (also plotted in Figure 8) has poor performance. Clearly, this is not surprising—one should not expect to see high performance in networks that are designed using simple heuristically designed and optimized models that reconcile robustness with the so-called "scale-free" models in the sense that they represent a highly engineered system like an ISP or the Internet as a whole. In contrast, we observe that even simple high-degree "hubs" that provide a relatively easy way to generate the connectivity hubs that provide a relatively easy way to generate the desired power law degree distribution. Given this insight, it is not surprising that theorists who consider probabilistic methods to generate "likely" graphs that make up much of the total configuration space are "likely" to find high-performance topologies, but these topologies are difficult to discern. Their network cores contain high connectivity hubs that are hallmarks of the degree-based models.

At the same time, these designs are highly "non-generic" and are difficult to discern. Their network cores contain high connectivity hubs that are hallmarks of the degree-based models. The fact that there are very few "likely" graphs that make up much of the total configuration space, and that a careful design process explicitly incorporating technological constraints can yield high-performance topologies, but these topologies are difficult to discern. Their network cores contain high connectivity hubs that are hallmarks of the degree-based models.

The current evidence suggests that high-performance networks are likely to be robust. However, it is also possible to identify probabilistically rare and poorly performing networks. This is a significant finding because it challenges the prevailing wisdom that high-performance networks are necessarily robust.
For every complex problem there is an answer that is clear, simple, and wrong.

*M.L. Mencken*

- Multiple models produce the same node-degree distribution
- They are VERY different
Section 2

COLD graphs
COLD \[BRB14\]

COLD = Combined Optimized Layered Design

- HOT pointed the way
  - but the optimisation model is not quite right
  - we really want to optimise actual costs
  - but costs in networks are complex
- Also, constraints in real networks are complex
  - traffic must be carried
  - ports limits must be respected
  - but real designs also include redundancy
- The resulting optimisation is too complex (for me)
- Tackle the problems in layers
  - the top-layer is \textit{inter-PoP}
    - PoP = Point of Presence
    - here we optimise simplified costs
  - second layer is between routers
    - here we build redundancy
COLD top layer optimisation

- Inter-PoP
  - recognise that long links cost more
  - so for the moment ignore links inside one city/PoP
  - avoids some (router) constraints
  - reduces the size of the problem to be tractable
    - optimisation is NP-hard

- Randomness
  - PoP locations (as in SERNs)
  - traffic matrix (as in HOT)
COLD costs

Link costs

\[ c_e = k_0 + k_1 \ell_e + k_2 \ell_e w_e, \]

where

\[ \ell_e = \text{the length of link } e, \]
\[ w_e = \text{the capacity of link } e, \]
\[ k_i = \text{a set of constants}. \]

Node cost = a “complexity” cost

\[ c_i = k_3 I(i \in N_H) \]

when node \( i \in N_H \) means it is a “hub” or a “core” node
Internet Topology Zoo

- We have a collection of (Internet) network topologies
  http://www.topology-zoo.org/
- One interesting thing is the variety of networks
  ▶ some look like Abilene
  ▶ others are hub-spoke networks
  ▶ others are more meshy
- The costs are flexible to allow all of these combinations
Optimisation is solved with a GA (Genetic Algorithm)
- we need to use a heuristic because the problem is NP-hard
- GA is still slowish, \( O(n^3) \)

Results
- are nicely tunable
- we can’t dispose of any of \( k_i \) (no simpler model)
- model parameters have operational meaning (costs)
- can match real variations of statistics

Matlab code
https://github.com/rhysbowden/COLD
Layering, hierarchy, redundancy and structure

- Now we need to build the router layer
  - incorporate redundancy
- But, often, lower layer is build with structure
  - hierarchy
  - graph operators

We’ll start talking about graph operators next
We haven’t worked out how to do estimation yet!
We haven’t worked out how to do estimation yet!
But I have a good idea how
Further reading


