Analyzing the Robustness of Graph Generators with the Infinity Mirror Test

Sal Aguinaga, Tim Weninger

University of Notre Dame

saguinag@nd.edu, tweninge@nd.edu

Introduction

Graph generators that learn a set of model parameters for any graphical representation of complex systems are not built free of bias and assumptions. A network's local and global structure used to derive such parameters can fail to capture crucial information necessary for synthetic graph generation. A simple recursive model-fitting to graphs derived from an original network can shed light on model degeneration and help us choose the right generator depending on the network property of interest.

We Propose

- *Infinity mirror test* for analysis of graph generator performance and robustness.
- A stress test that operates by recursively fitting a model to itself.
- A comprehensive evaluation of network properties as measured on the original graph

Complex Networks

Transportion

Commu-nications

Brain

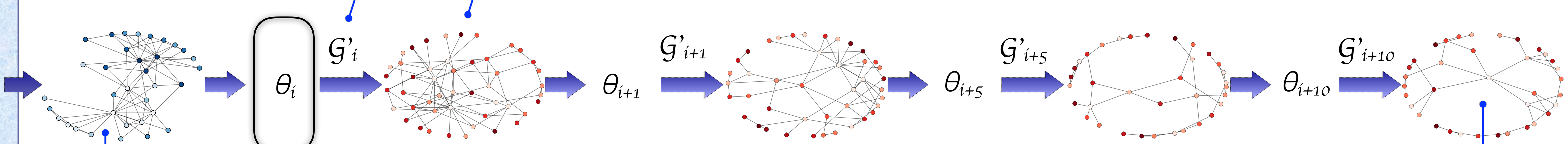
Social

Analyze the robustness and stability of graph generators

generate a synthetic graph using model θ_i

metrics: compute correlation distance, centralities; we then compare these properties to those of the reference graph

Original Graph



Given a graphical representation of any complex network or complex system

A model θ_i of the input graph is learned using one of the following graph generators: **Chun-Lu, BTER, Kronecker**

after the **tenth** recurrence

Example shown here is the karate club graph using the Kronecker product graph generator.

Generators

The graph generators examined:

- Kronecker Product
- Chung-Lu: optimized versions
- Exponential Random Graph*
- Block Two-Level Erdos-Renyi

Datasets

- C. elegans neural(269/2,965)
- Power Grid (4941/6,594)
- ArXiv GR-QC(5,242/14,496)
- Internet Routers(6,474/13,895)
- Enron emails(36,692/183,831)
- DBLP(317,080/1,049,866)

* Results not shown for exponential random graph model due to consistent model degeneracy

Experiments

We computed the following metrics:

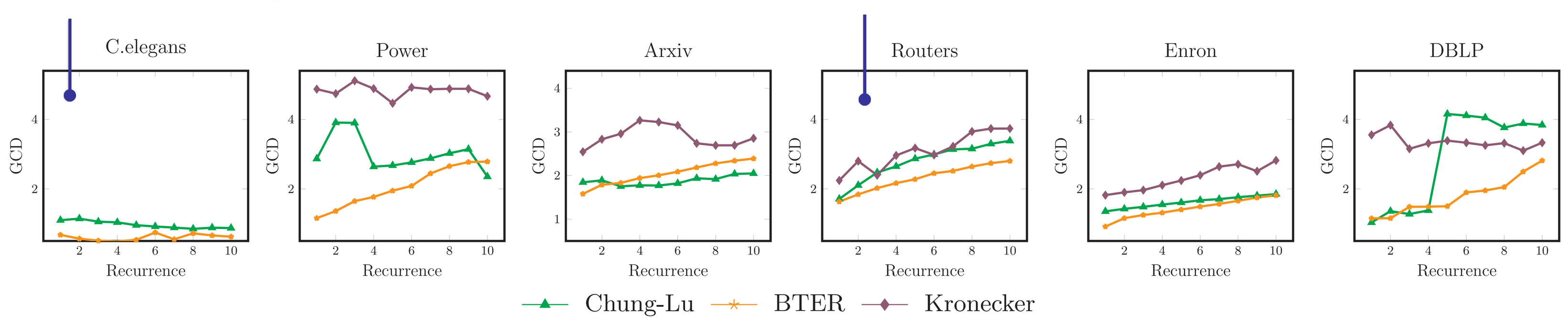
- Graphlet correlation distance
- Eigenvector centrality (not shown)
- Hop-plot
- Degree distribution
- Clustering Coefficients
- Assortativity

Results

Graphlet correlation distance

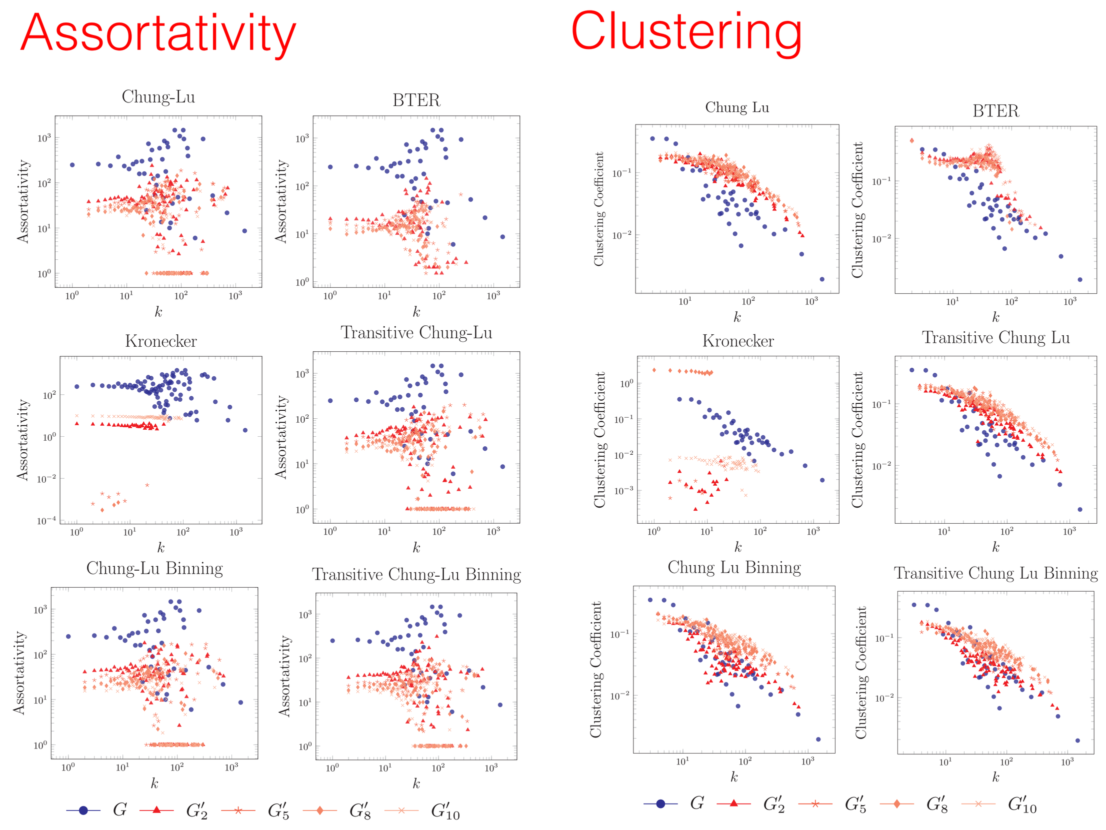
Kronecker cannot learn a model on graphs that don't follow power-law degree distribution

In general, graphlet correlation distances increase as the recurrence number increases



Assortativity

Clustering



Assortativity and clustering computed on Internet Routers dataset.

Chung-Lu models degenerate as the recurrence # increases

BTER model tends to shift assotativity signature downwards

Kronecker model yields assortativity signatures with linear characteristics especially as the recurrence # increases

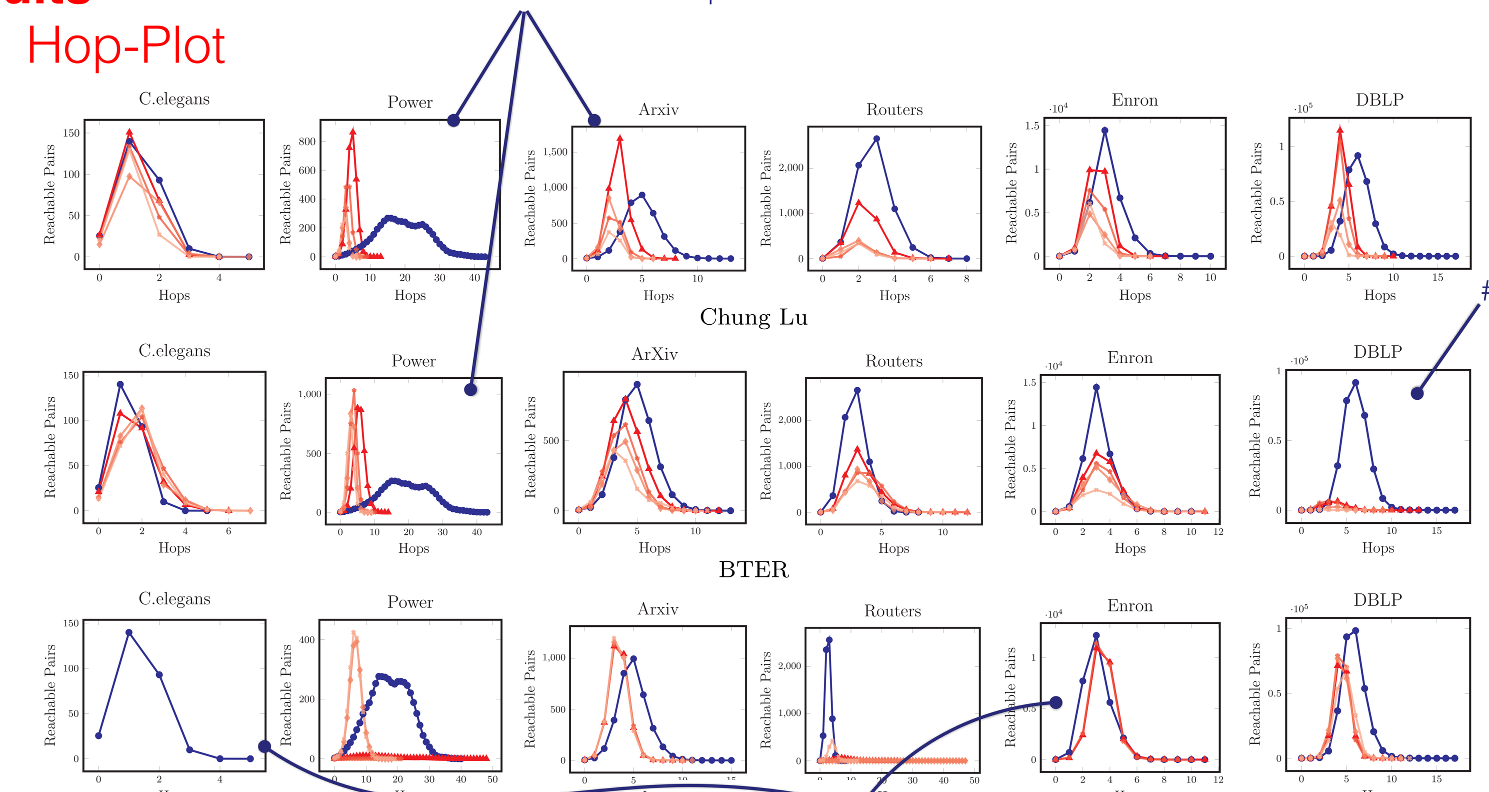
On clustering, Chung-Lu models do well, but Kronecker and BTER hold different signatures

Results from ERGM are omitted due to model degeneracy

Results

Hop-Plot

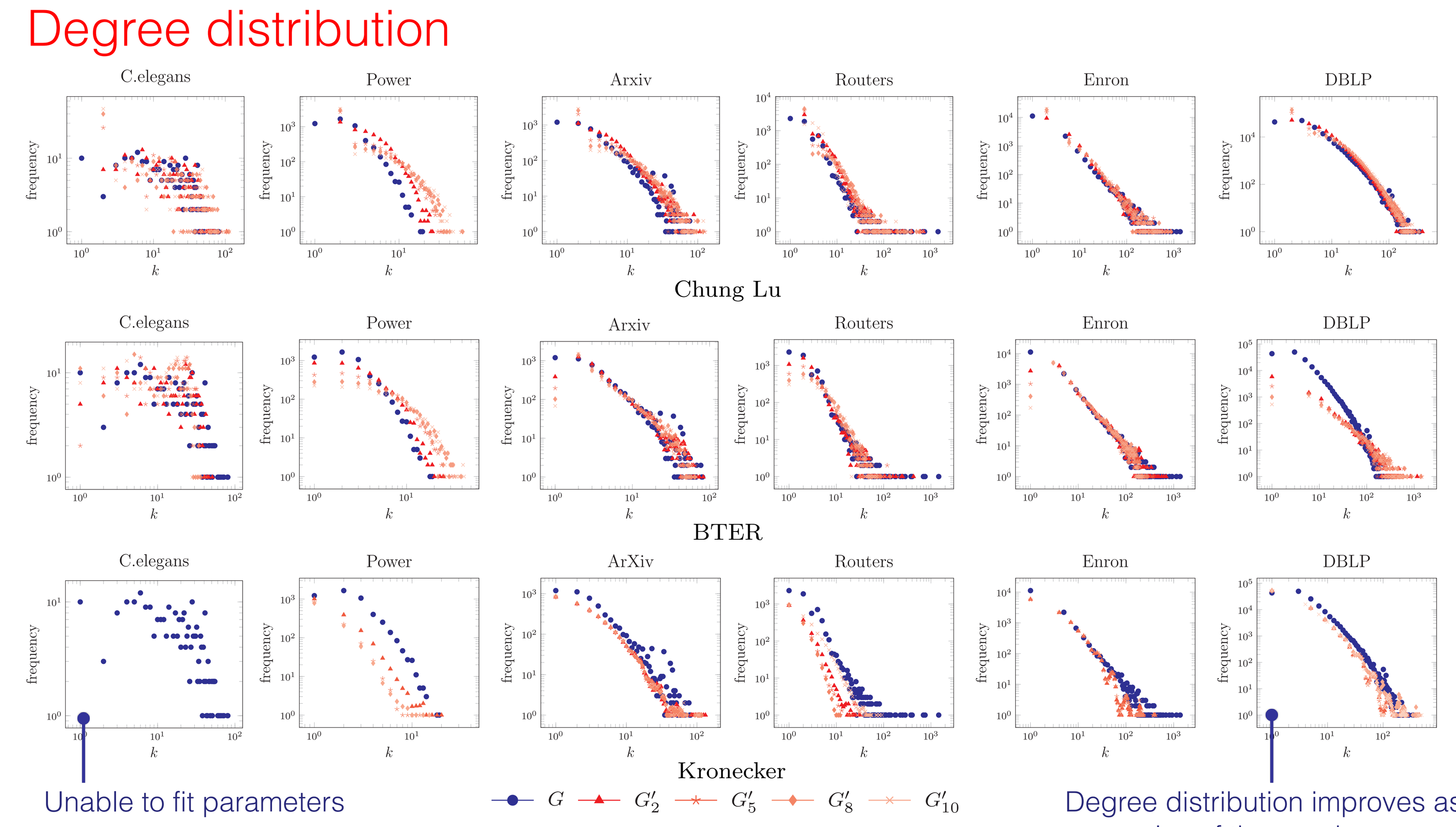
more reachable pairs as the recurrence # increases



number of reachable pairs plummet

Model degenerates or is unable to learn a model

Degree distribution



Unable to fit parameters

Degree distribution improves as the size of the graph grows

Conclusions

Graph generators like Kronecker, Chung-Lu and its variants, or BTER can be used to learn and fit model parameters on any input graph and in turn use the model to generate reasonable synthetic graphs. The strength and robustness in the model can be assessed using a simple recursive test we call *infinity mirror*. The test can help us choose the best generator to fit the task.

REFERENCES

T. G. Kolda, et al., A scalable generative graph model with community structure. SIAM Journal on Scientific Computing, 36(5), 2014.

J. Leskovec, et al., Graphs over time: densification laws, shrinking diameters and possible explanations, SIGKDD, 2005.

S. Mussmann, et al. Incorporating assortativity and degree dependence into scalable network models, AAAI 2015.

This work is supported by the Templeton Foundation under grant FP053369-M/O. Copyright held by authors. 12th International Workshop on Mining and Learning with Graphs, San Francisco, CA. 2016