Generating Networks by Learning Hyperedge Replacement Grammars

Sal Aguiñaga

November 15, 2017

Weninger Lab & iCeNSA
Department of Computer Science and Engineering
University of Notre Dame, Notre Dame, Indiana

Table of contents

1. Learning Hyperedge Replacement Grammars
   Experimental Results
2. Model Robustness
3. Tree Decomposition
4. Summary
5. Collaborations
6. Future Directions

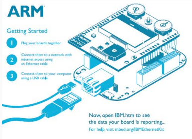
Introduction

Summary Network modeling is critical to the study of complex systems. It enables researchers to examine emergent behavior and related phenomena arising from local milieu. The mechanism and function of this local area or micro structures is an area that remains nebulous. Learning what the underlying patterns are and how they function in real world representations of complex systems (networks) is critical to the evolution of scientific tools required for today's data-saturated environment.

Why is this important?

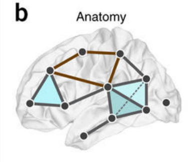
- Medir [para modelar], predecir y controlar los sistemas complejos, Albert-László Barabási at CCS17 Cancun, Mexico.
- To answer "questions about the sustainability of modern socio-ecological system", Deep Time Frontiers of Ecological Networks, Jennifer A. Dune, at NetSci 2017, Indianapolis, IN

Motivation

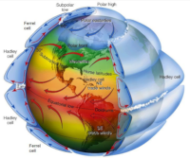


Internet of Things[1]

Human Genome[2]



Brain networks[3]



World weather [4]

[1]IOT `http://iotcentral.io`
[2]Gene banks: identifying relationships between genes, `pubgene.com`
[3]Discovering how dynamic brain connections give rise to thoughts and behaviors, Bassett and Sporns Nature Neuroscience 20, 353364 (2017)
[4]Prescient questions in climate change `tutiempo.net`

Thesis Statement

In this thesis,

I investigate the relationship between graph theory and formal language theory that allows for a Hypererdge Replacement Grammars (HRG) to be extracted from any connected graph and learn the building blocks of real world graphs.

I propose, develop and evaluate algorithms that leverage the generating properties of HRGs to solve problems in graph mining and network science in general.

Background and definitions

What is . . .

Network science **Network models** **Hyperedge Replacement Grammar**

Background

What is ...

Network science
Study of complex networks drawing on theories and methods in graph theory, statistical mechanics, data mining, ...

Ex:
· telecomm nets
· biological nets
· cognitive and semantic nets

Network models
In understanding the interactions in real-world networks models provide the foundation necessary.
Ex:
· Random graph generation models yield structures for comparing real-world networks.

Hyperedge Replacement Grammar

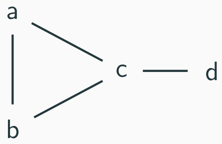
- Hyperedge replacement is a fundamental technique for graph and hypergraph rewriting[a]
- Context-free rewriting formalism for graph generation.[b]

[a]Drewes et al. (1997)
[b]Chiang et al. (2013)

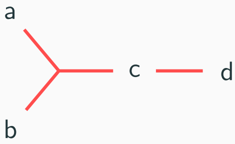
Background - Graphs, hypergraphs, hyperedges, ...

What is a graph, what is hypergraph, or a hyperedge

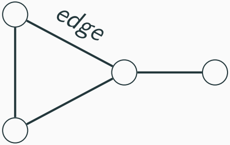
A hypergraph extends a Graph abstraction in the sense that edges are allowed to connect to an arbitrary number of vertices (not just two).

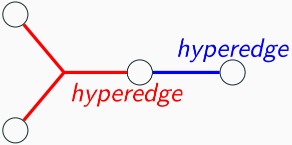


Graph



Hypergraph (H)





Background - HRGs

What is a hyperedge replacement grammar

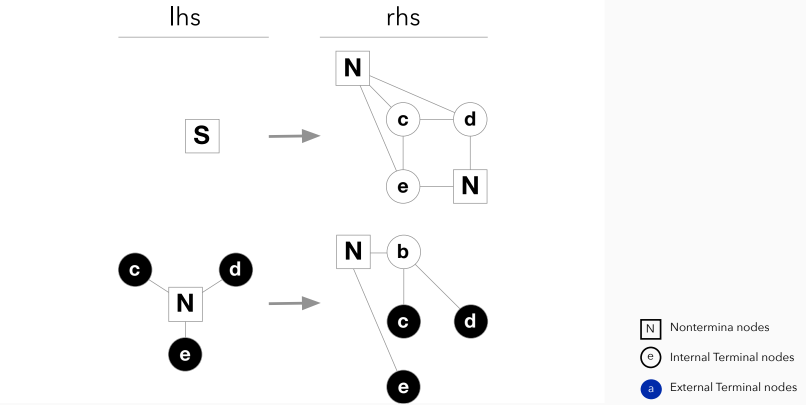
An HRG is graph rewriting system. It represents the instructions on how the graph is pieced together.

A hyperedge replacement grammar is a tuple $\mathcal{G} = \langle N, T, S, \mathcal{P} \rangle$

N	finite set of nonterminal symbols.
T	finite set of terminal symbols.
S	distinguished starting nonterminal, where $S \in N$
P	finite set of production rules $A \rightarrow R$
	$LHS \rightarrow RHS$

Background - HRG continued

An HRG $\mathcal{G} = \langle N, T, S, \mathcal{P} \rangle$. Ex: Production rules where $A \rightarrow R$ denotes that we will replace A with R. Shown below is an example of left hand side (lhs) and right hand side (rhs) elements for two productions $\in \mathcal{P}$.



Background and definitions

Network science **Network models**

Hyperedge Replacement Grammar

The focus of my work is on HRG graph models.

Related work - Network models

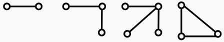
Other approaches used in the field ...

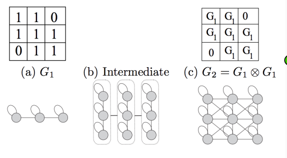
Random graph model	$G(N, p)$ model wires pairs of nodes with probability p	Erdős and Rényi (1960); Gilbert (1959)
Preferential attachment	Captures hub formation in the graph; a direct extension of random graph model	Barabási and Albert (1999)
Stochastic block model	In sociometry, the objective is to break up a graph into groups or blocks. These models helps in finding communities.	Girvan and Newman (2002)

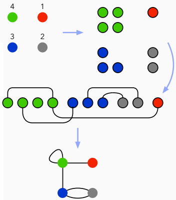
Related work - network properties based models

Other modeling approaches relying on the properties of network in question . . .

$$\Pr(\mathbf{X} = \mathbf{x}) = \frac{\exp\{\boldsymbol{\theta}'\mathbf{z}(\mathbf{x})\}}{\kappa(\boldsymbol{\theta})},$$







Exponential Random Graph Model

Generates graphs with similar # of triangles or wedges as observed empirically

Kronecker product

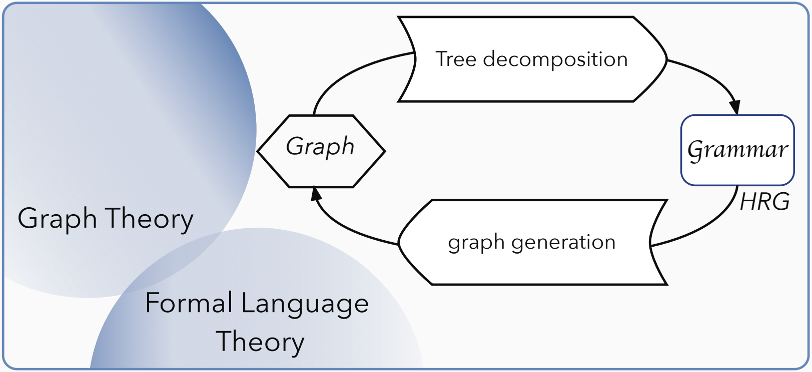
Infers a model as a 2x2 or 3x3 matrix, then uses matrix multiplication to grow graphs

Chung and Lu

Generates graphs using degree sequence

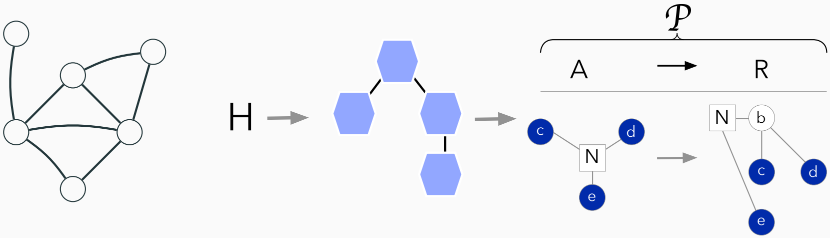
Learning Hyperedge Replacement Grammars

Model inference and graph generation

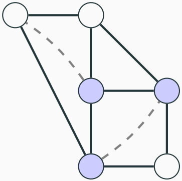


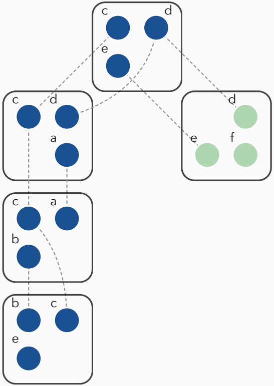
HRG Model Inference

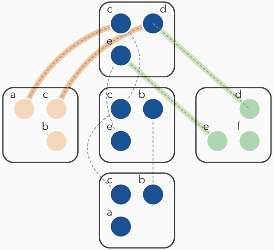
Here is how the HRG model works



Maximum Cardinality Search is the algorithm used in our tree decomposition step

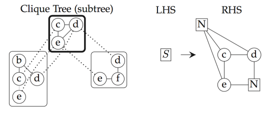


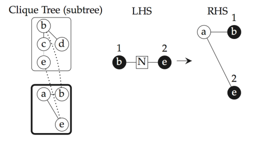


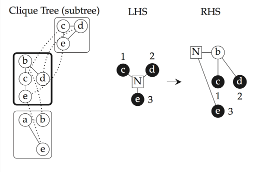


HRG extraction

Extracting an HRG for







Rule 1

. . . the root

Rule 2

. . . a leaf[5]

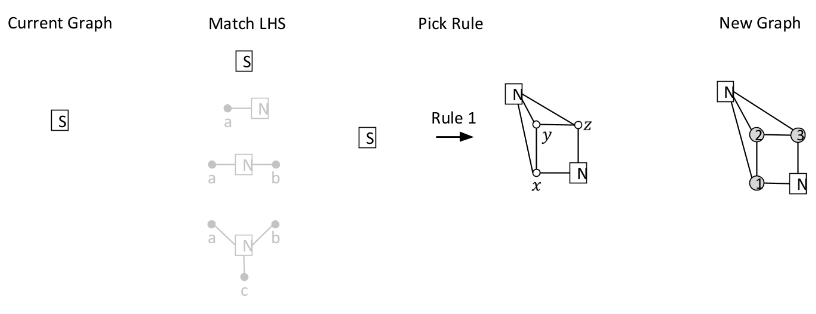
Rule 3

. . . a middle clique

[5] No Nonterminal symbols on *rhs*

Graph generation - Stochastic

Generating a synthetic graph



From production rules to graph generation

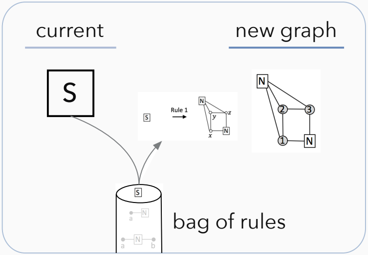
Graph generation options

Isomorphic graphs	Interesting, loss-less, but impractical (not used)
Stochastic	Automatically grows graphs of approximate size and we **throw away bad graphs** that don't match the input
Fixed size	We discuss this one later

Stochastic graph generation - characteristics

Stochastic

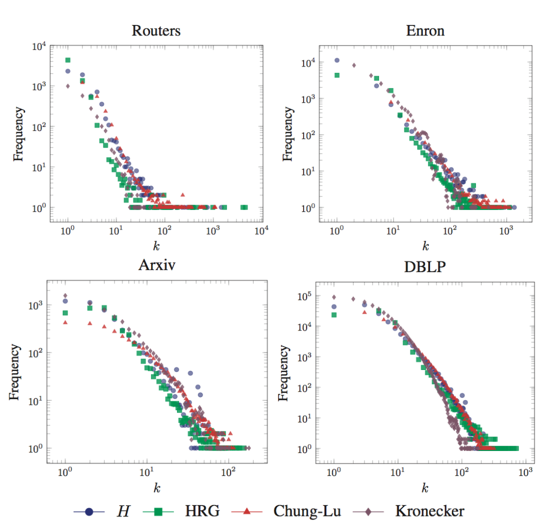
- On large graphs we learn an HRG by sampling 2-3 (300 node) subgraphs
- PHRG: identical rules in $(A \rightarrow R)$ are merged and we keep a count that allows us to define a probability distribution over the rules



Experimental results - Stochastic Generation

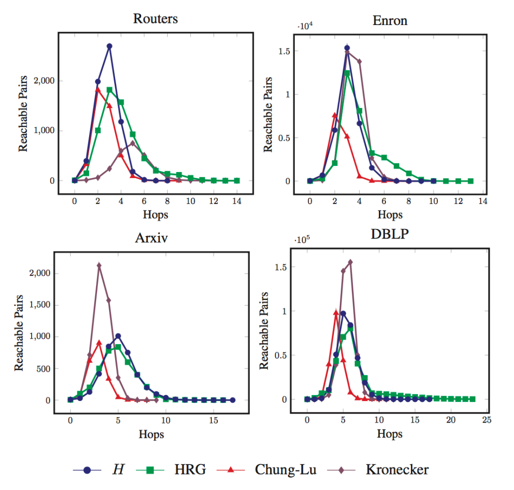
Degree distribution

Evaluated HRG on a diverse array of public datasets



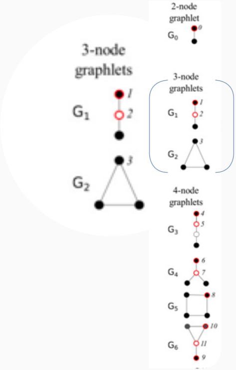
Experimental results - Stochastic Generation

Hop Plot



Graphlet correlation distance

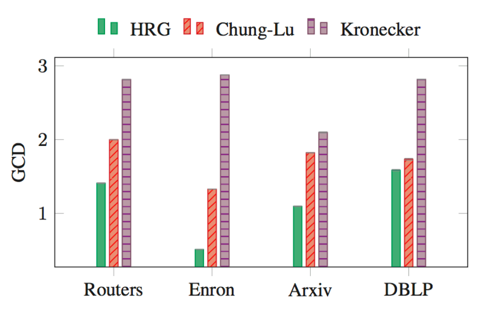
Graphlet correlation distance (GCD[6])



- A new network metric for network alignment
- Compares two networks by considering graphlet structures at the node level

[6] Yaveroğlu et al. (2015)

Graphlet Correlation Distance

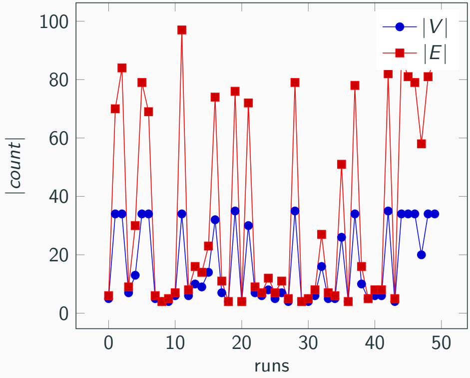


NB: lower is better.

A Significant Limitation of the HRG Model

We discovered a limitation

- Stochastic generation yielded graphs of various sizes. The median-size seemed to be close to the empirical size, but the size-variance is impractical.



Example where the reference graph has 34 nodes and 78 edges (50 runs)

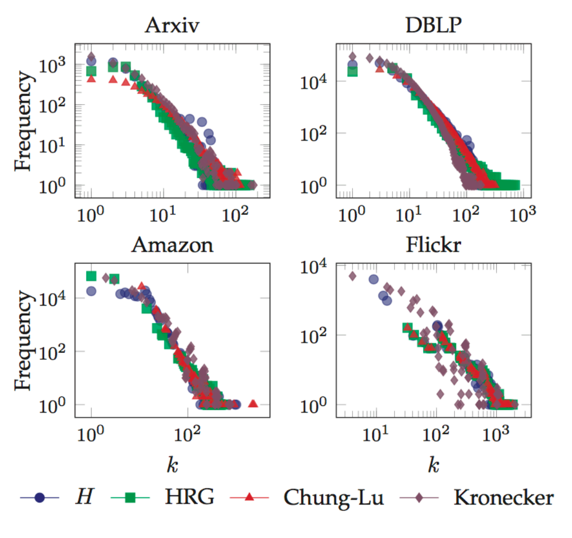
Fixed-Size addresses the limitation of Stochastic generation

PHRG defines a probability distribution over rules. $P(H^*)$,[7] so we want to sample from $P(H^* \mid \|H^*\| = n)$	We pre-compute a table of rule firing probabilities α [8]	We select a path through α that generates a graph of specified size

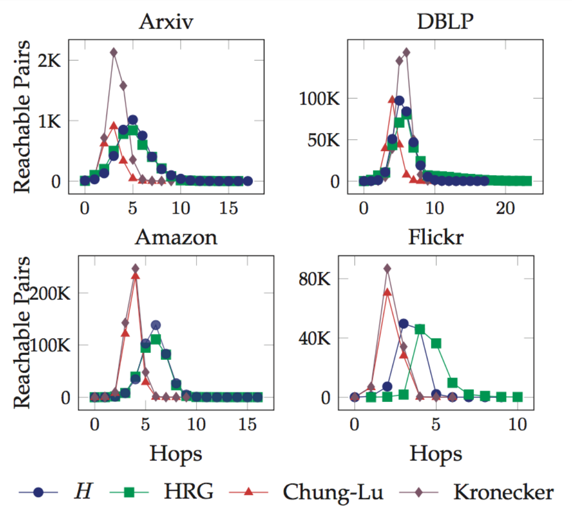
Once we have computed the α, we can sample a graph of size n

[7] H is the original graph and H^* is the synthetic graph
[8] Code by Chiang

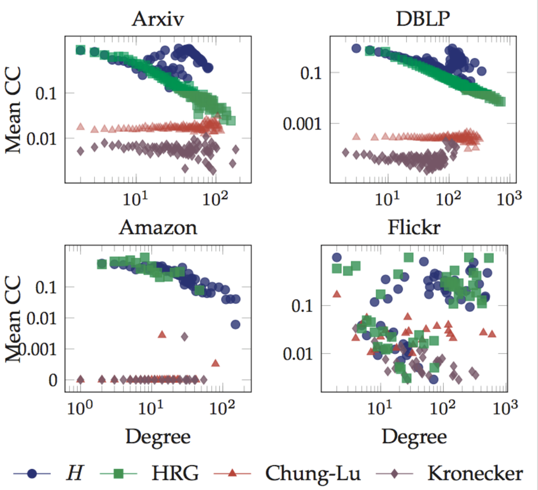
Network Statistics - Degree distribution



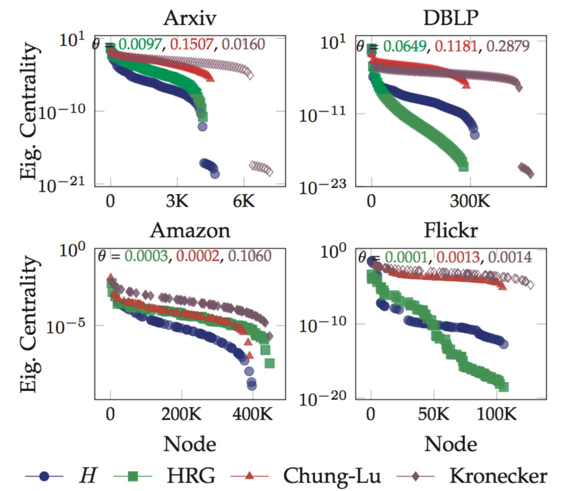
Network Statistics- Hop-Plot



Network Statistics - Clustering Coefficients



Network Statistics - Eigenvector Centrality



Network Statistics - Graphlet counts and correlation distance

Table 1: Motif Statistics and GCD

Graphs									GCD
Routers	13511	1397413	9863	304478	6266541	177475	194533149	18615590	0
HRG	13928	1387388	9997	288664	6223500	174787	208588200	18398430	1.41
Kronecker	144	61406	0	80	10676	973	642676	551496	2.81
Chung-Lu	4787	356897	6268	81403	1651445	13116	35296782	4992714	2.00
Enron	727044	23385761	2341639	22478442	375691411	6758870	4479591993	1371828K	0
HRG	79131	4430783	49355	554240	13123350	556760	688165900	54040090	0.51
Kronecker	2598	5745412	1	1011	608566	49869	1.89468000	141065K	2.88
Chung-Lu	322352	23590260	1191770	16267140	342570000	10195620	3967912K	2170161K	1.33
arXiv	89287	558179	320385	635143	4686232	382032	11898620	7947374	0
HRG	88108	606999	320039	656554	5200392	455516	15691941	9162859	1.10
Kronecker	436	224916	1	293	47239	4277	3280822	2993351	2.10
Chung-Lu	927	232276	6	967	87868	11395	2503333	3936998	1.82

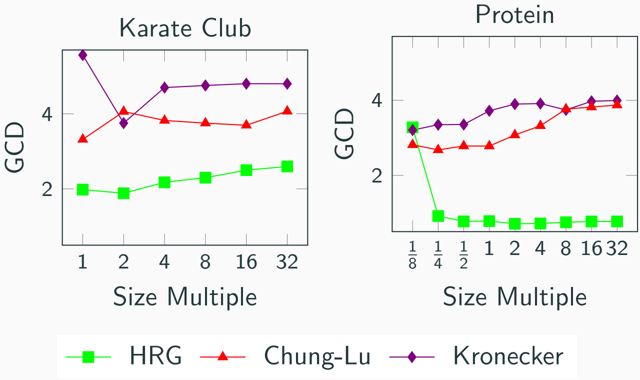
GCD - computes statistics on a per node basis

Motif counting - a more traditional global metric

Graph Extrapolation

Given a small snapshot of the network

Could we use our models to infer larger (or smaller) network with the same local and global properties?



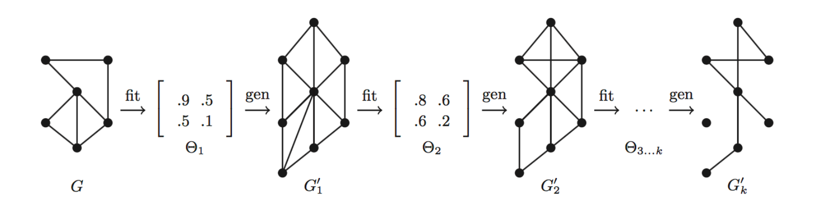
HRG generate good results at small size multiples & GCD scores remain mostly leveled as the size multiple grows.

Model Robustness

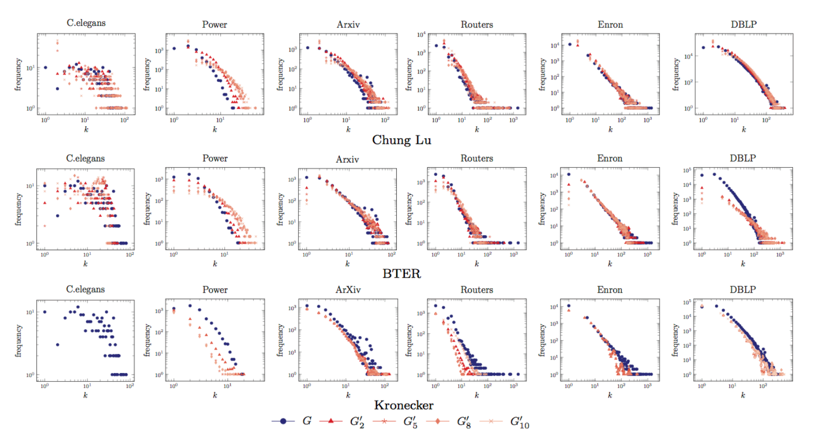
Infinity Mirror Test

Simple test aims to test how well a graph model captures the important features in the empirical graph.

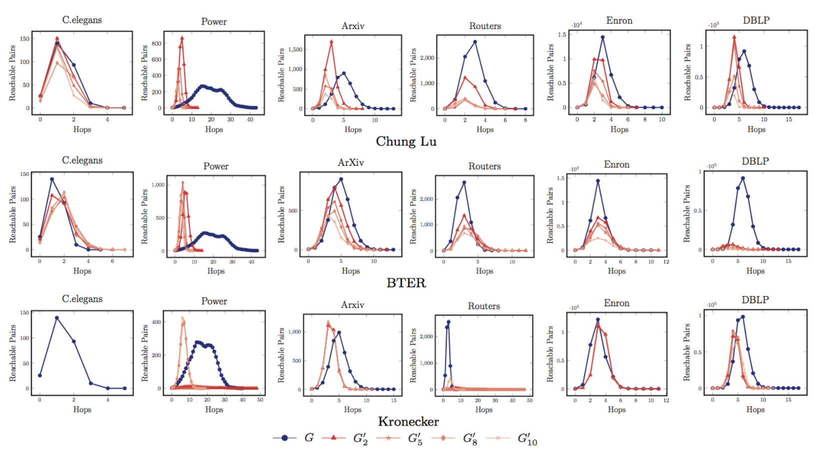
Example infinity mirror test on the Kronecker model



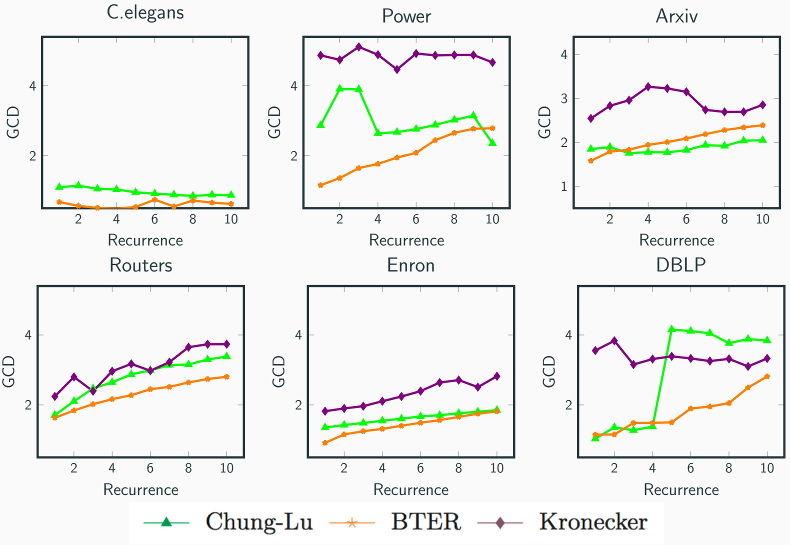
Network Statistics- Degree distribution



Network Statistics- Hop-plot

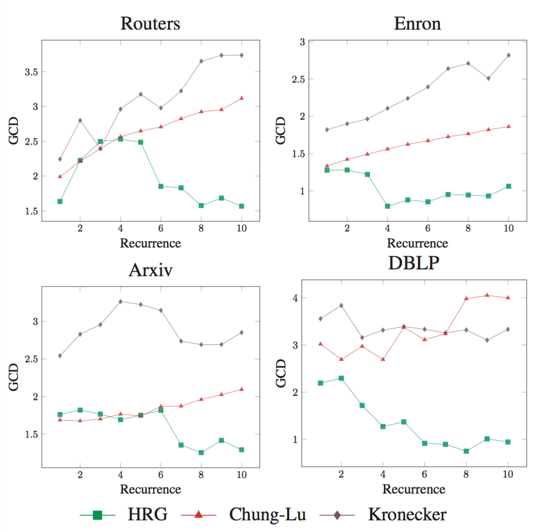


Graphlet Correlation Distance



Infinity Mirror Test on HRG Graphs

GCD - Notice the downward trend



Tree Decomposition

Tree Decomposition Effect on Production Rules

Does choice of tree decomposition affect the production rules?

- What kind of production rules result given the choice of *TD* algorithm?
  - $\longrightarrow$ We explore six algorithms and examine the rules.
  - $\longrightarrow$ We find a rule set from intersection of the production rules that result from multiple tree decompositions.

Tree decomposition algorithms

Table 2: We identified these *TD* algorithms for this study.

TD algorithm	Description
MCS	Maximum cardinality search is a simple heuristic
MCSM	Minimum triangulation extension to MCS.
MinD	Minimum degree is a well known general-purpose ordering scheme and is widely used in sparse matrix computation.
MinF	Minimum fill consists of greedy node elimination with the fewest edges are added breaking ties arbitrarily.
Lex-M	Derived from lexicographic breadth-first search for minimal triangulation.
MMD	Multiple minimum degree

Tree Decomposition - Results

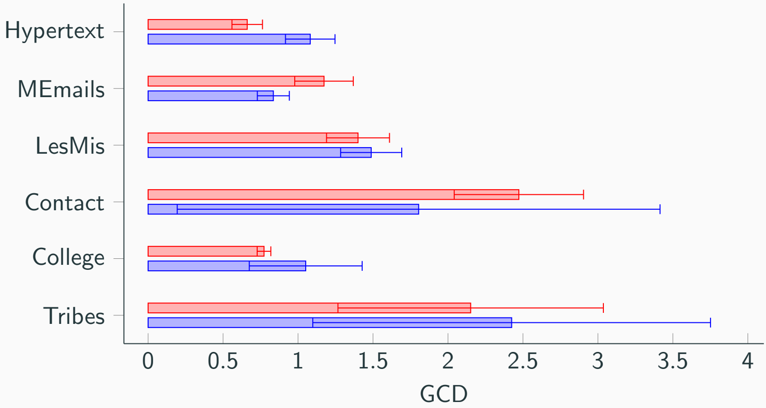
Width of the tree as a function of *TD* algorithm

Dataset	The width of the tree						
	mcs[9] μ (σ)	mcs	lexm	mcsm	mind	minf	mmd
LesMis	9 (0)	11	9	11	9	9	9
contact	40	42	43	43	50	40	40
arenas-jazz	59 (0)	88	77	81	104	59	73
pdzbase	6	9	12	13	6	6	6
ucforum	126	326	361	341	282	276	279
Hypertext	76 (0)	80	89	89	76	76	76
Infectious	39 (0)	65	56	128	42	40	49
emailEuCore	34 (0)	41	46	45	35	34	35
EuroRoad	6.6 (2.6)	42	30	48	19	16	16
College Msg	87.6 (20.3)	459	602	543	404	394	403

[9] HRG uses maximum cardinality search

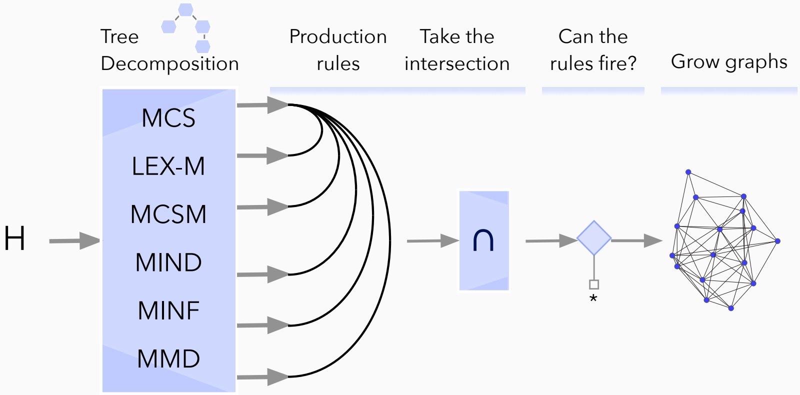
More Results - Graphlet Correlation Distance

■ MCS (baseline) and ■ MinF produce the best results. Below is the GCD computed for graphs generated using productions derived from the these two *TD* algorithms.



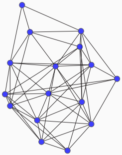
Tree Decomposition - Methods

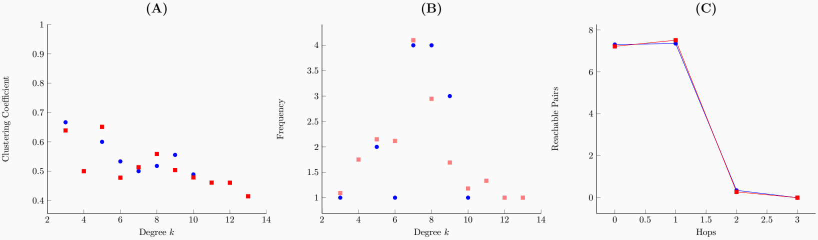
Six different *TD* algorithms, then we find the overlap and isomorphic intersection.



From the isomorphic intersection of rules

Graph: Social network of tribes of the Gahuku-Gama





(A) Clustering Coefficients, (B) Degree Distribution, and (C) Hop Plot

Summary

Summary

1. HRG for general graphs

- Model inference
- Stochastic Graph generation.

2. PHRG & Fixed size

- Probabilistic HRG
- Fixed-size graph generation

3. Model Robustness

- Infinity Mirror Test
- Graph size extrapolation

4. Tree Decomposition

- Exploring tree decomposition effects on HRG graphs
- Isomorphic intersection of the production rules grow HRG graphs

Collaborations

Collaborations

Temporal HRG (THRG)

An extension of HRG to time-stamped dynamic graphs. We explored temporal dynamics to enhance HRG. Our objective was to capture latent features during network growth Pennycuff et al. (2017).

Latent variable HRG (LaTHRG)

Addressing a limitation in HRG graph generation step that selects rules based on frequency rather than their place in terms of tree-height Wang et al. (201x).

Future Directions

Future Direction

Dig deeper & extend these concepts to other areas

Neural network architectures

Sinha et al. (2017); Negrinho and Gordon (2017)

Zoph and Le (2016)

Analytical Methods

Newman (2002); Leskovec et al. (2010)

Graph Contractions

Bernstein et al. (2017); Wang et al. (2014)

Thank you.

Special thanks to each member of my thesis committee for lending me their time and effort.

Prof. Tim Weninger

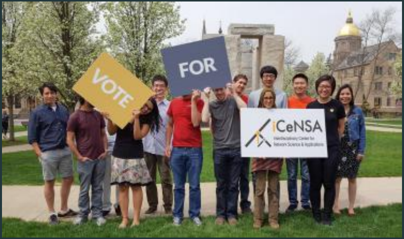
Advisor

Prof. James Evans

Prof. David Chiang

Prof. Nitesh Chawla

I would like to acknowledge my lab mates for lending me an ear and discussing with me all kinds of crazy ideas!



The Weninger Lab & iCeNSa

Acknowledgements

Templeton Foundation (FP053369-M/O)

NSF IIS (#1652492)

Software

`github.com/nddsg/PHRG`

Download source code and collaborate

`twitter.com/abitofalchemy`

Follow my work and tell me what you think.

These slides are based on:

`github.com/matze/mtheme`

References

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439):509–512.

Bernstein, A., Däubel, K., Disser, Y., Klimm, M., Mütze, T., and Smolny, F. (2017). Distance-preserving graph contractions. *arXiv preprint arXiv:1705.04544*.

Chiang, D., Andreas, J., Bauer, D., Hermann, K. M., Jones, B., and Knight, K. (2013). Parsing graphs with hyperedge replacement grammars. In *ACL (1)*, pages 924–932.

Drewes, F., Kreowski, H.-J., and Habel, A. (1997). Hyperedge replacement, graph grammars. *Handbook of Graph Grammars*, 1:95–162.

Erdős, P. and Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61.

Gilbert, E. N. (1959). Random graphs. *Ann. Math. Statist.*, 30(4):1141–1144.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Ghahramani, Z. (2010). Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042.

Negrinho, R. and Gordon, G. (2017). Deeparchitect: Automatically designing and training deep architectures. *arXiv preprint arXiv:1704.08792*.

Newman, M. E. (2002). Random graphs as models of networks. *arXiv preprint cond-mat/0202208*.

Pennycuff, C., Aguinaga, S., and Weninger, T. (2017). A temporal tree decomposition for generating temporal graphs.

Sinha, A., Sarkar, M., Mukherjee, A., and Krishnamurthy, B. (2017). Introspection: Accelerating neural network training by learning weight evolution. *arXiv preprint arXiv:1704.04959*.

Wang, C. X., Aguinaga, S., Weninger, T., and Chiang, D. (201x). Growing better graphs with latent-variable probabilistic graph grammars. *TBD*.

Wang, T., Lang, C., and Feng, S. (2014). Joint tree of combinatorial maps. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–28. Springer.

Yaveroğlu, Ö. N., Milenković, T., and Pržulj, N. (2015). Proper evaluation of alignment-free network comparison methods. *Bioinformatics*, 31(16):2697–2704.

Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.