Reproducing Network Structure: a Comparative Study of Random Graph Generators

Mikhail Drobyshevskiy, Denis Turdakov and Sergey Kuznetsov

Ivannikov Institute for System Programming of the RAS, Moscow, Russia

November 30, 2017

Agenda

Motivation for random graph generators and our goal

Method and models

Experiments

Agenda

Motivation for random graph generators and our goal

Method and models

Experiments

Motivation for Random Graph Generators (RGG)

Network science:

- various graph domains: biological, social, citations, lexical, mobile calls, infrastructure, autonomous systems, etc
- many network mining tasks and network mining tools (e.g. community detection)

Problems solved with help of RGGs:

- significance testing
- scalability testing
- data anonymization

Classical RGG approach: extract graph properties, develop mathematical model which guarantees them

Figure: scale-free

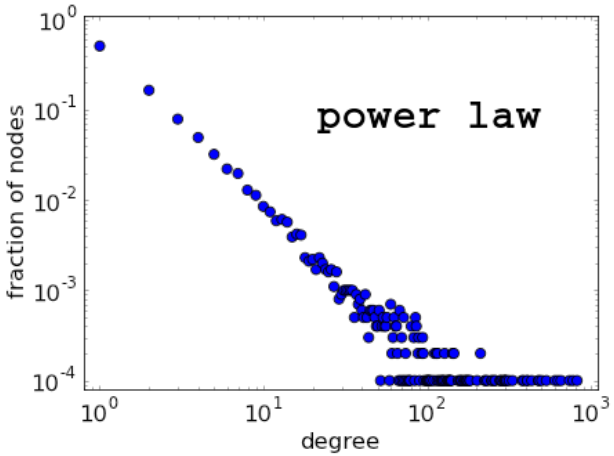
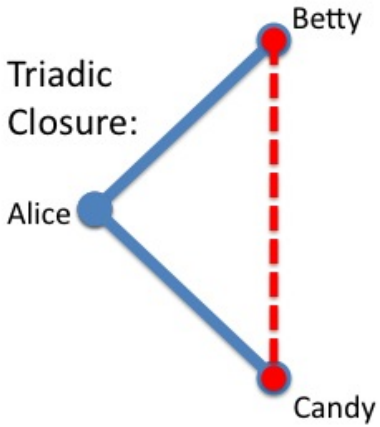


Figure: high clustering



Problems of RGGs:

- although graph domains share some graph properties, each domain has its own specific ones
- no exhaustive list of all graph properties, some may be unknown

Our goal

Of our interest: universal generators of directed graphs similar to a given one

Good universal RGG follow 2 requirements on generated graphs:

1. *Similarity* to the original one in terms of manifold graph metrics
2. *Variability* wide enough to mimic natural diversity across a graph domain

Comparative study plan:

- select state-of-the-art universal RGGs
- compare them over various domains in terms of graph properties capturing capabilities

Agenda

Motivation for random graph generators and our goal

Method and models

Experiments

Method

Similarity and variability estimation

- no universal graph similarity metric
- compare over several graph metrics (numeric and distributional)

Metrics:

1. Degree: degree distribution, assortativity
2. Triads: clustering coefficient, subgraph distribution
3. Distance: diameter, hop-plot

Dataset: 8 different domains, 1 graph per domain, moderate size (1K–100K nodes)

Models selection

Criteria:

1. Model can be applied to an arbitrary directed graph without fitting its parameters
2. Availability of the algorithm implementation

Selected models:

1. (2010) Stochastic Kronecker Graphs (SKG)[1]
2. (2016) GScaler[2]
3. (2017) Embedding based Random Graph Generator (ERGG)[3]

---

[1] J. Leskovec et al. "Kronecker graphs: An approach to modeling networks," Journal of Machine Learning Research, 11(Feb):985–1042, 2010

[2] J. Zhang and Y. Tay, "Gscaler: Synthetically scaling a given graph," in EDBT, 2016, pp. 53–64

[3] M. Drobyshevskiy, A. Korshunov, and D. Turdakov, "Learning and scaling directed networks via graph embedding," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer International Publishing, 2017

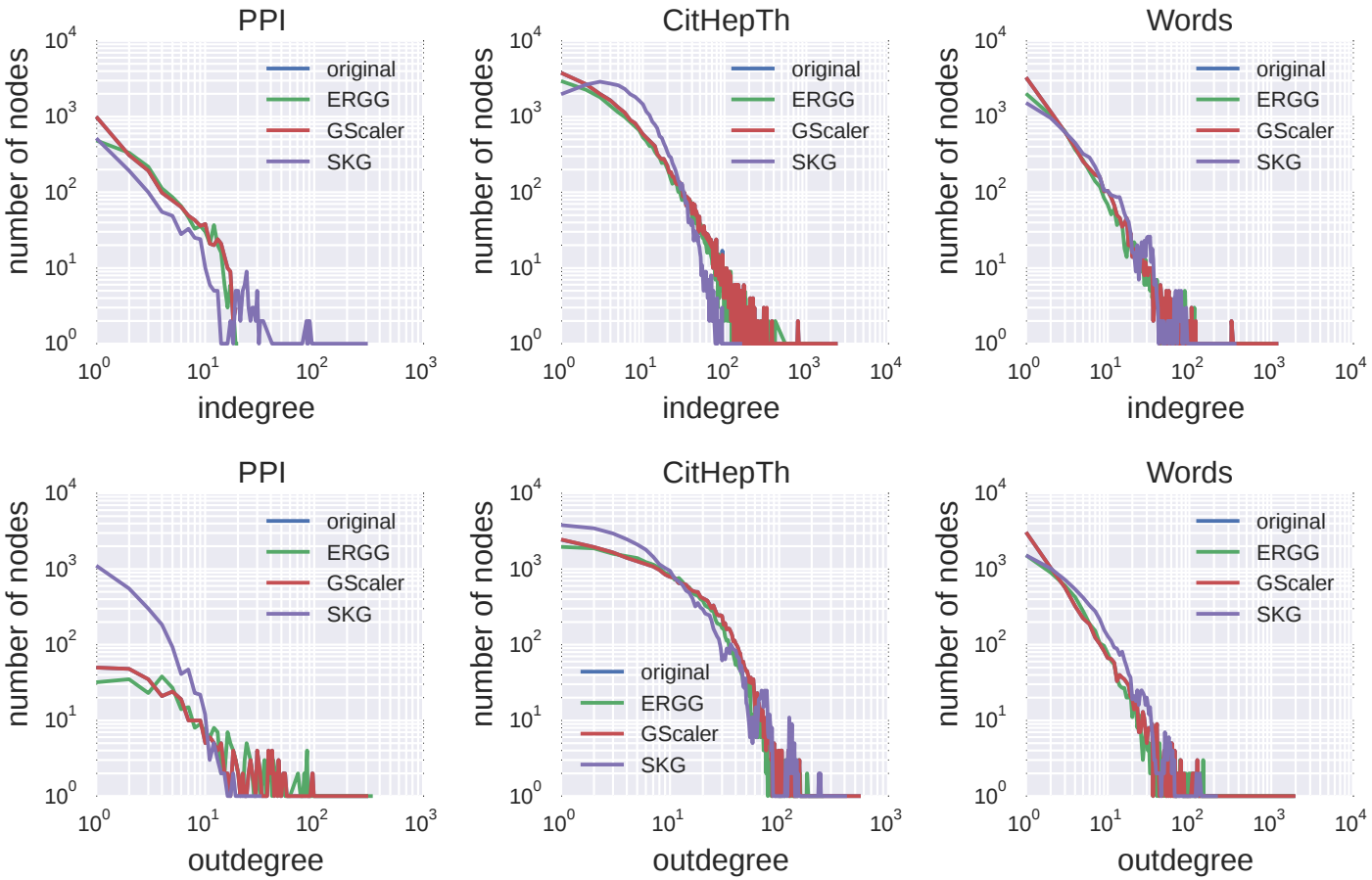
Agenda

Motivation for random graph generators and our goal

Method and models

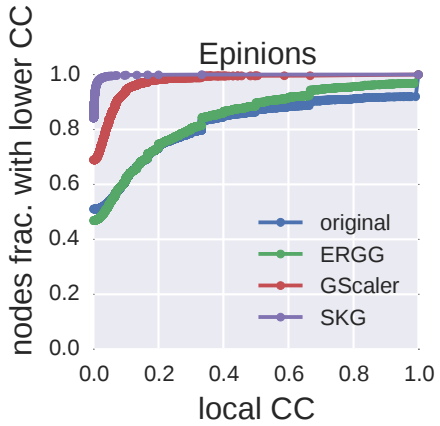
Experiments

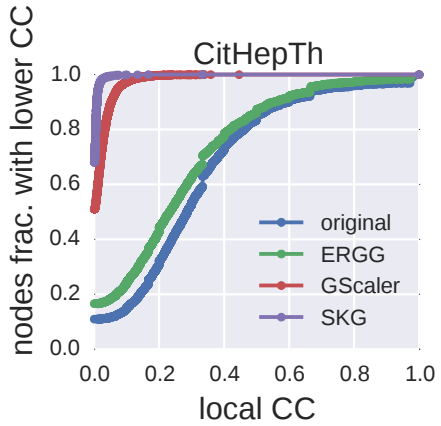
Degree distribution

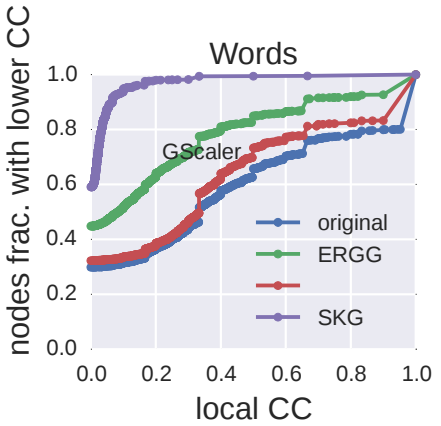


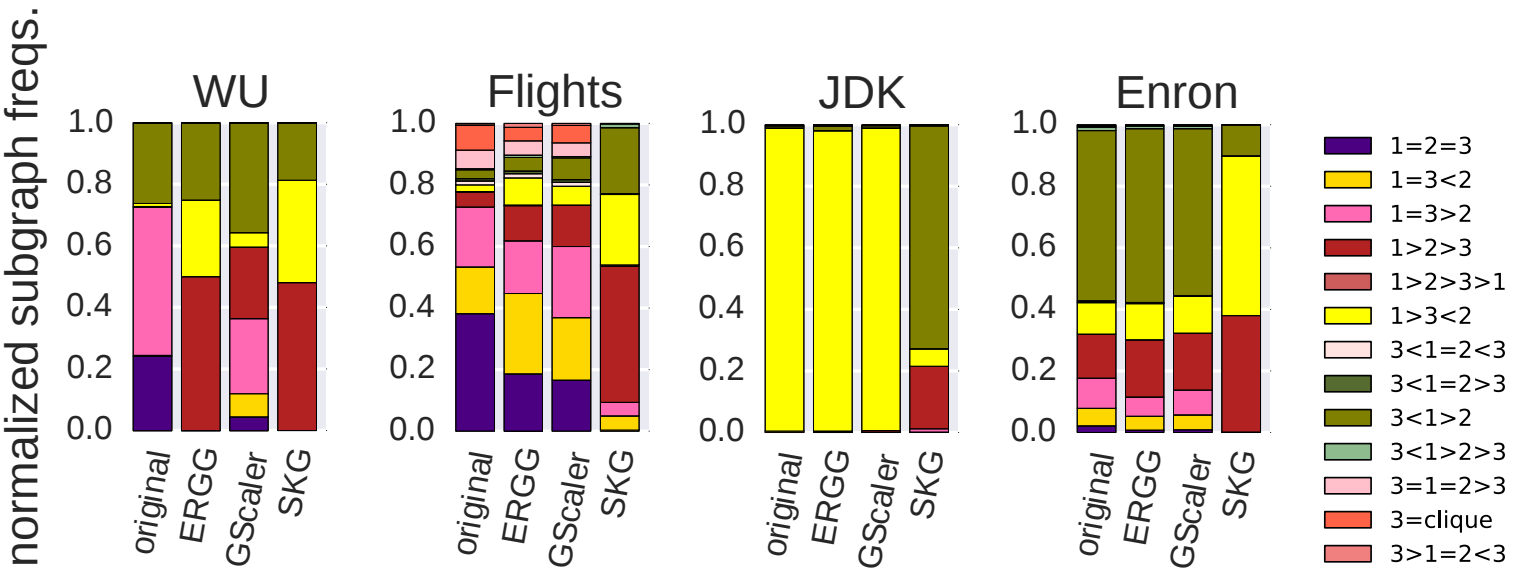
Gscaler repeats DD almost perfectly, while ERGG reproduces form of DD not exactly but much closer to the original than SKG

Triads





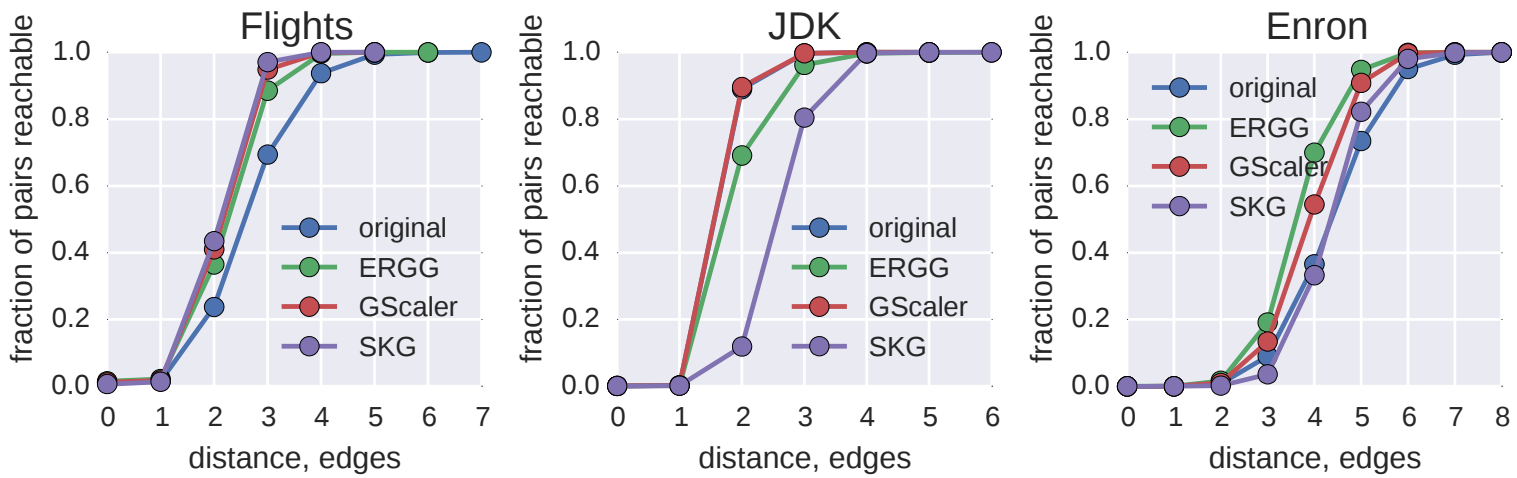




SKG fails to capture high clustering, while Gscaler's and ERGG's results diverge much across domains

3-subgraphs are captured by ERGG and Gscaler in almost all domains

Distance



Similarity of hop-plots of generated graphs to the originals also varies for different domains

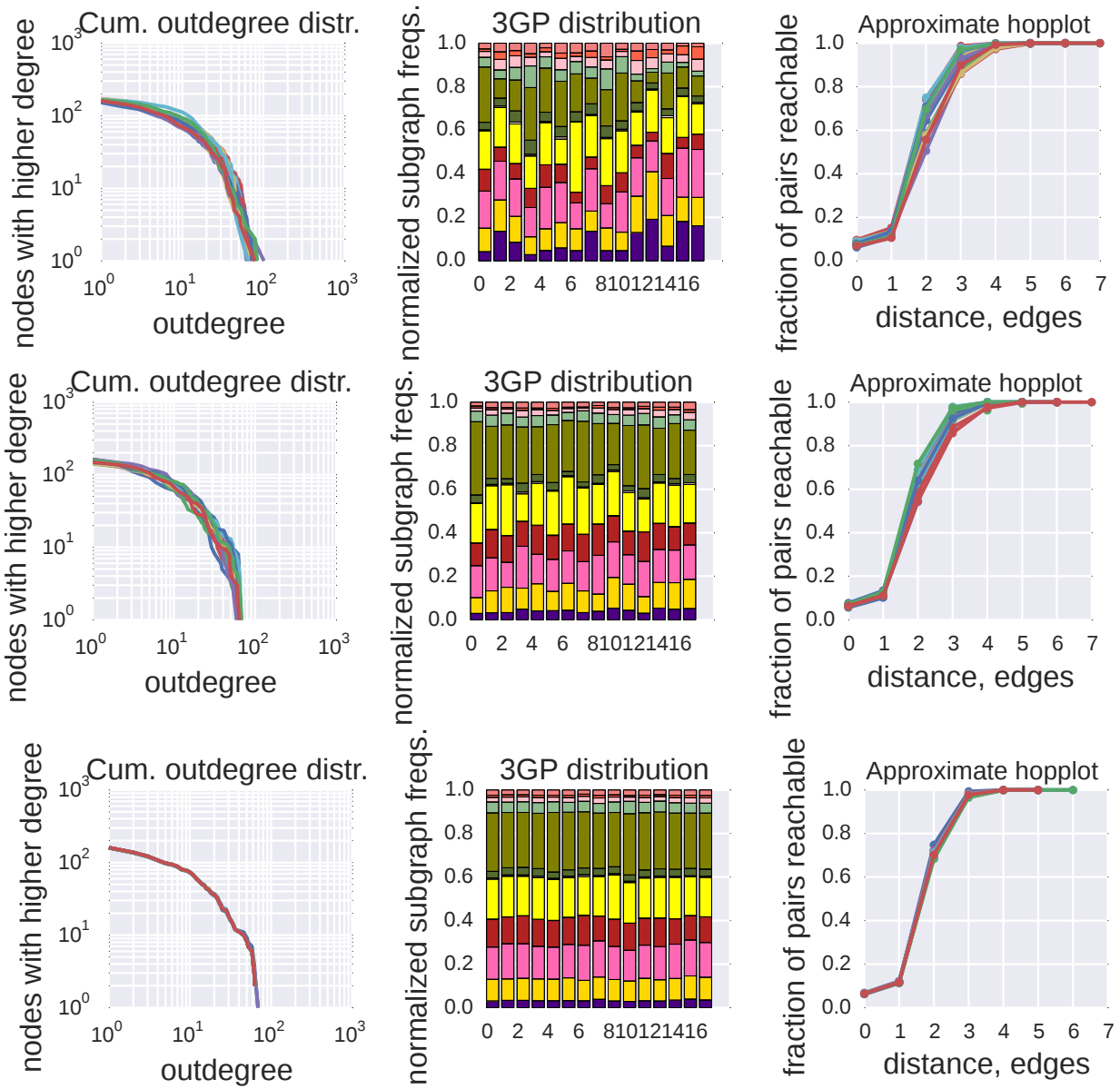
Closest hop-plot reproduction corresponds to the closest CC reproductions

Variability

top original variability

middle ERGG: similar variability in in-, out-DD, hop-plot, CC distribution and lower variance in 3-subgraph distribution

bottom GScaler: close to zero variability in DDs, 3-subgraph, and hop-plot



Conclusions

Work done

- Analysed the capability of RGGs (ERGG, Gscaler and SKG) to imitate a given graph from an *arbitrary* domain
- Compared *similarity* and *variability* of generated graphs in terms of various graph metrics

Resume

- SKG graphs are more similar to each other than to original rgaphs
- ERGG and Gscaler capture most of tested graph properties, although their accuracy varies at different domains

Practical recommendations

- Apply Gscaler for very close graph imitating, especially when degree correlations matter
- Apply ERGG for domain emulation and synthetic dataset representativity

Thanks for attention!

Dataset for similarity tests

domain, subdomain	**graph**	**nodes**	**edges**
bio, protein-protein interactions	PPI[4]	2239	6452
social, trust network	Epinion[4]	49288	487183
citations	CitHepTh[4]	27770	352807
lexical, word adjacency	Words[5]	7381	46281
mobile calls	WU[6]	72146	100974
infrastructure	Airport[4]	1574	28236
autonomous system, software dependency	JDK[4]	6434	53892
social collaboration, emails	Enron[4]	87273	321918

[4] http://konect.uni-koblenz.de/networks

[5] http://www.weizmann.ac.il/mcb/UriAlon/sites/mcb.UriAlon/files/uploads/CollectionsOfComplexNetwroks/darwinbookinter_st.txt

[6] http://www.pnas.org/content/suppl/2010/10/15/1013140107.DCSupplemental/SD02.txt

Dataset for variability tests

Collection of twitter-ego networks[7].
Selected were 15 graphs close in number of nodes ($|N| \in [170; 180]$) and number of edges ($|E| \in [2000; 3000]$).

[7] http://snap.stanford.edu/data/egonets-Twitter.html