Sampling Large Complex Networks via Maximal Entropy Random Walk

Joon-Hyuk Ko[1] Sung Hoon Kim[2]

[1]Dept. of Physics Astronomy
Seoul National University
jhko725@snu.ac.kr

[2]Dept. of Physics Astronomy
Seoul National University
josephkim11@snu.ac.kr

December 5, 2019

Table of Contents

Table of Contents

Status Quo: Real World Graphs are Getting Bigger!

With graphs growing ever larger, it is more and more difficult to parse an entire graph and its properties:

Name	Nodes	Edges	Description
YahooWeb	1,413 M	6,636 M	WWW pages in 2002
LinkedIn	7.5 M	58 M	person-person in 2006
	4.4 M	27 M	person-person in 2005
	1.6 M	6.8 M	person-person in 2004
	85 K	230 K	person-person in 2003
Wikipedia	3.5 M	42 M	doc-doc in 2007/02
	3 M	35 M	doc-doc in 2006/09
	1.6 M	18.5 M	doc-doc in 2005/11
Kronecker	177 K	1,977 M	synthetic
	120 K	1,145 M	synthetic
	59 K	282 M	synthetic
	19 K	40 M	synthetic
DBLP	471 K	112 K	document-document
flickr	404 K	2.1 M	person-person
Epinions	75 K	508 K	who trusts whom

Size of various real world graphs (Kang et al[1])

[1] U Kang, C. E. Tsourakakis, and C. Faloutsos, "PEGASUS: A Peta-Scale Graph Mining System - Implementations and Observations", Proc. 9th IEEE International Conference on Data Mining, 2009.

Status Quo: Real World Graphs are Getting Bigger!

With graphs growing ever larger, it is more and more difficult to parse an entire graph and its properties:

Two different solutions to the problem:

1. Direct distributed computation of global graph properties via MapReduce[1]
2. Sampling parts of the graph to infer its properties[2]

- Graph sampling is advantageous in situations where one does not have dedicated resources to run large scale algorithms on.
- Useful in exploratory analysis of large graphs, where one has yet to decide whether to invest resources to the project or not.

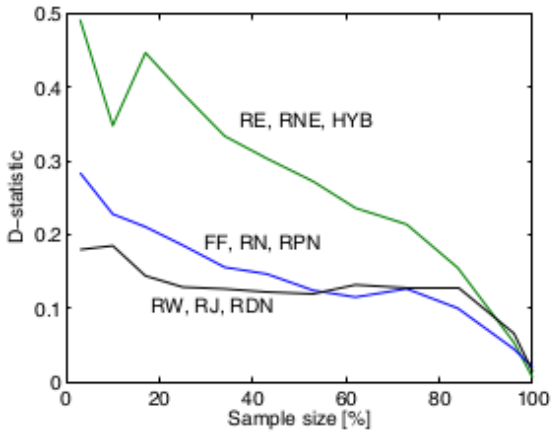
Key point of graph sampling: choice of the sampling algorithm

[2] V. Krishnamurthy et al, "Reducing Large Internet Topologies for Faster Simulations", NETWORKING 2005.

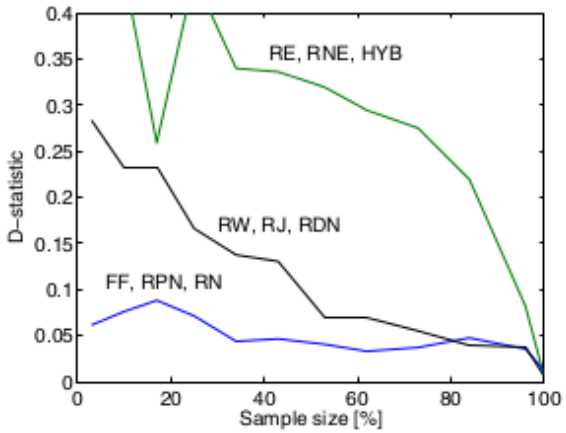
What is Good Sampling?

"Goodness" of a graph sample: ill-defined

In Leskovec et al[3], two different goals are presented: "back-in-time" and "scale-down"



(a) Scale-down



(b) Back-in-time

Lower D-statistic indicates a better sample. The different initials denote different sampling algorithms.

A "good" sampling algorithm depends on the sampling goal!

[3] J. Leskovec and C. Faloutsos, "Sampling from Large Graphs", Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.

Sampling Goals and Algorithm Performance

Note that even for the same "scale-down" goal, the "best performing algorithm" differs for each graph property.

	Static graph patterns								Temporal graph patterns				
	in-deg	out-deg	wcc	scc	hops	sng-val	sng-vec	clust	diam	cc-sz	sng-val	clust	**AVG**
RN	0.084	0.145	0.814	0.193	0.231	0.079	0.112	0.327	0.074	0.570	0.263	0.371	0.272
RPN	**0.062**	**0.097**	0.792	0.194	**0.200**	0.048	0.081	0.243	0.051	0.475	0.162	0.249	0.221
RDN	0.110	0.128	0.818	0.193	0.238	0.041	0.048	0.256	0.052	0.440	**0.097**	0.242	0.222
RE	0.216	0.305	**0.367**	0.206	0.509	0.169	0.192	0.525	0.164	0.659	0.355	0.729	0.366
RNE	0.277	0.404	0.390	0.224	0.702	0.255	0.273	0.709	0.370	0.771	0.215	0.733	0.444
HYB	0.273	0.394	0.386	0.224	0.683	0.240	0.251	0.670	0.331	0.748	0.256	0.765	0.435
RNN	0.179	0.014	0.581	0.206	0.252	0.060	0.255	0.398	0.058	0.463	0.200	0.433	0.258
RJ	0.132	0.151	0.771	0.215	0.264	0.076	0.143	**0.235**	0.122	0.492	0.161	**0.214**	0.248
RW	0.082	0.131	0.685	0.194	0.243	0.049	**0.033**	**0.243**	**0.036**	**0.423**	**0.086**	0.224	**0.202**
FF	0.082	0.105	0.664	0.194	**0.203**	**0.038**	0.092	**0.244**	0.053	0.434	0.140	**0.211**	**0.205**

Table 1: Scale-down sampling criteria. On average RW and FF perform best.

Data from Leskovec et al[3]. Rows correspond to different sampling algorithms and smaller numbers indicate good performance

To find a "good sampling algorithm", we must decide our graph parameter of interest!

Our Objective: Node Centrality

Node centrality

- Measure of importance of a node in a graph
- Degree centrality, Betweeness centrality, Closeness centrality

Eigenvector centrality

- Meaure of "influence" of a node in a graph
- Google PageRank, Katz centrality, ..
- 'node connected to important nodes is important'
- useful for weighted graphs, signed graphs
- biological implication : important parameter for temporal brain network analysis
- exact definition : $x_i \leftarrow$ component of the first eigenvector

$$A\mathbf{x} = \lambda\mathbf{x}$$

Algorithm of Interest: Maximal Entropy Sampling

Let $\gamma_{i_0 i_t}^{(t)}$ be a random walk trajectory of length t, passing through nodes $(i_0, i_1, ..., i_t)$
Probability of moving along this trajectory:

$$P(\gamma_{i_0 i_t}^{(t)}) = P_{i_0 i_1} P_{i_1 i_2} \cdots P_{i_{t-1} i_t} \quad \text{(Markovian Property)}$$

In general, the probability depends on all the intermediate nodes - two different paths from i_0 to i_t with the same length t, have different probabilities

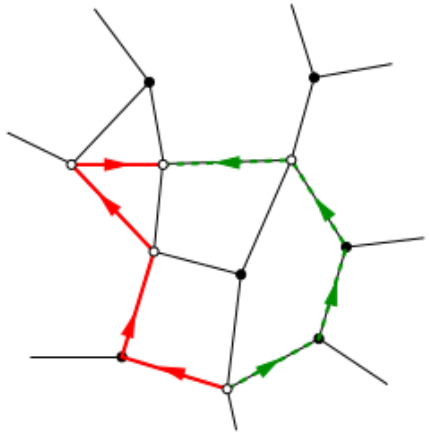
But is there a "special" type of random walk that allows the two paths to have the same probabilities?

Yes! $\rightarrow$ Maximal Entropy Random Walk[4]

[4] Z. Burda et al, "Localization of the Maximal Entropy Random Walk", Phys. Rev. Lett. **102**, 160602, 2009.

Algorithm of Interest: Maximal Entropy Random Walk

Consider the following figure[5]:



For simple random walk, the probabilities for each path are given by:

$$P_{red} = \frac{1}{4} \cdot \frac{1}{3} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{192}; \quad P_{green} = \frac{1}{4} \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{144}$$

For maximal entropy random walk, the two paths have the same length, and hence, the same probabilities.

[5] Z. Burda et al, "The Various Facets of Random Walk Entropy", arXiv cond-mat.stat-mech 1004.3667 , 2010.

Algorithm of Interest: Maximal Entropy Random Walk

In physics, maximal entropy random walk(MERW) is interesting because it corresponds to the "path integral formalism of quantum mechanics"

In data mining, MERW is interesting because the steady state probability of node i is given by

$$\pi_i^* = \psi_i^2$$

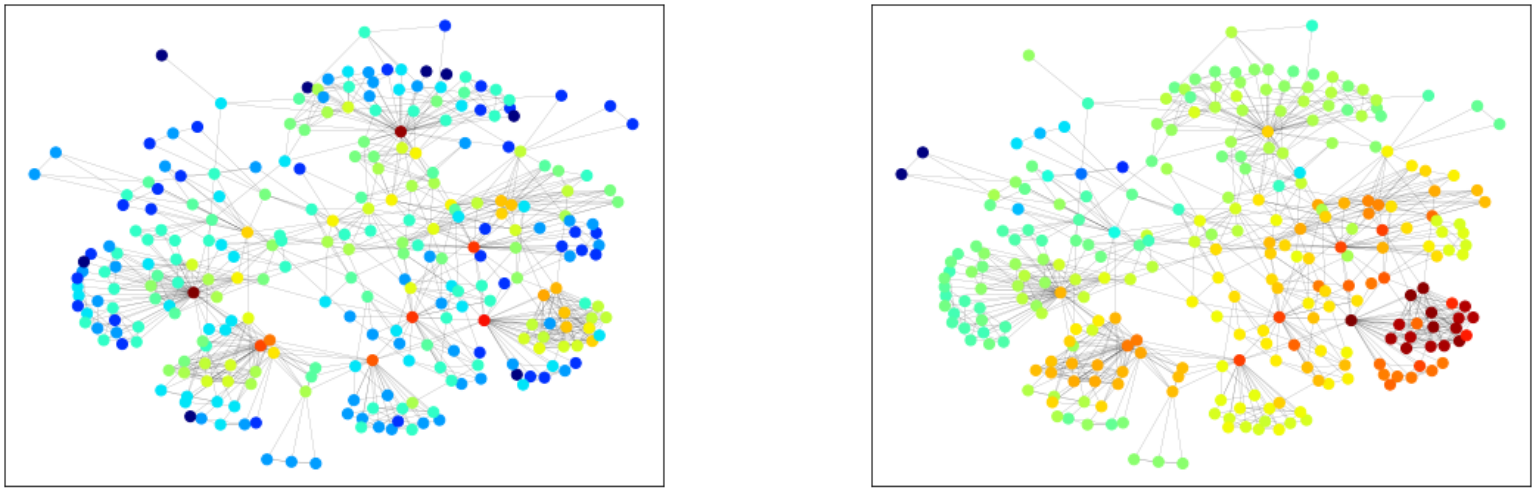
(ψ_i: i-th element of the principal eigenvector of the adjacency matrix)

ψ_i is the eigenvector centrality of the ith node!

MERW is a "promising" algorithm for accurately sampling node centrality!

Algorithm of Interest: Maximal Entropy Random Walk

Graph: a subgraph of the arXiv cond-mat dataset



Steady state distributions of General Random Walk(left) and Maximal Entropy Random Walk(right)

Note the difference in the stationary distributions!

Problem Definition

- GIVEN: A large graph to sample, with the assumption that the full adjacency matrix is not available
- IMPLEMENT: A new physics-based sampling algorithm - Maximal Entropy Random Walk Sampling
- ANALYZE: Basic benchmarks of the new sampler, with respect to other similar algorithms such as Forest Fire, or Simple Random Walk.
- VERIFY: Whether the MERW sampler shows superior performance in sampling nodes with high eigenvector centrality

Table of Contents

Sampling Algorithms for Comparison

Graph sampling algorithms can be classified into two main classes[6]:

1. Global information based sampling: random node/edge sampling, PageRank-based node sampling, etc.
2. Traversal based sampling: random walk-based sampling, Forest-Fire sampling, etc.

In our problem, we assume the full adjacency matrix is not available $\rightarrow$ only traversal based sampling algorithms are allowed!

$\therefore$ Compare against commonly used traversal based sampling algorithms:

- General Random Walk Sampling
- Metropolis-Hastings Random Walk (Uniform Node Random Walk) Sampling
- Forest Fire Sampling

[6] P. Hu and W. C. Lau, "A Survey and Taxonomy of Graph Sampling", arXiv cs.SI 1308.5865, 2013.

Datasets used for Algorithm Evaluation

Both synthetic and real world dataset are used:

Dataset Name	Nodes	Edges	Specification
Barabási–Albert	10000	99945	Generated graph using Barabási–Albert model
ca-CondMat	21363	91342	arXiv Condensed Matter collaboration network
AS	6474	13895	Network of Internet routers(Autonomous Systems)
Erdős–Rényi	10000	99952	Generated graph using Erdős–Rényi model
RoadNet	20010	27038	Road network of California

Table 1: Specifications of Dataset

Implementation of MERW Sampling

Originally, the transition probabilities for MERW require the principal eigenvalue of the adjacency matrix: contradicts our problem definition

In Sinatra et al[7], a local version of the algorithm is derived, and the transition probability from node i to j is given by:

$$p_{i \to j} \propto d_j^{1-\nu}; \quad d_i \propto d_{nn,i}^{-\nu}$$

(ν: characteristic exponent of the degree correlation,
$d_{nn,i}$: average degree of the nearest neighboring nodes of node i)

ν is not a given parameter - must find a way to approximate it!

[7] R. Sinatra et al, "Maximal-Entropy Random Walks in Complex Networks with Limited Information", Phys. Rev. E, **83**, 030103, 2011.

Implementation of MERW Sampling

The algorithm was implemented using Python 3.7.

1. Initially, $\nu \leftarrow 0$, and the random walker is left to equilibrate for t_0 steps. During this time, sample is not taken, and only $\log(d_i)$ and $\log(d_{nn,i})$ is recorded for each node the random walker visits.
2. After t_0 steps, linear regression is performed on the saved list of $\log(d_i)$ and $\log(d_{nn,i})$ to find ν.
3. The random walker transitions from node i to j with $p_{i \to j} \propto d_j^{1-\nu}$, and as it visits new nodes, ν is updated using recursive linear regression[8]. This is repeated until the desired number of nodes have been sampled.

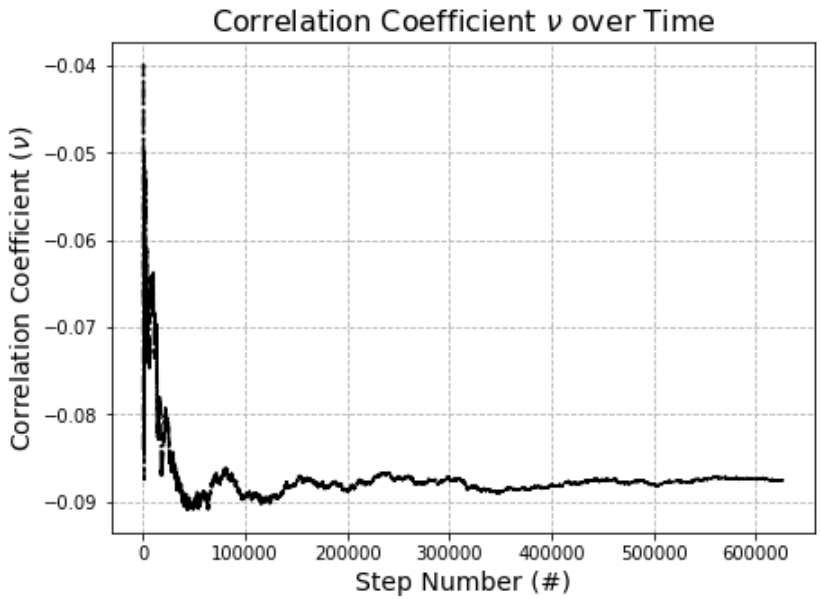
[8] J. H. Klotz, "Updating Simple Linear Regression", Statistica Sinica **5**, 1, 1995.

Implementation of MERW Sampling

Algorithm 1 Maximal Entropy Random Walk Sampling

Input: $G(V, E), V_0 \subset V$, constants r, t, t_0
Output: $G_s(V_s, E_s)$

```
function MERW(w_i, ν)
    w_f ← v with probability ∝ d_v^{1−ν} for all v ∈ N(w_i)
    return w_f
end function
function CalcNu(R)
    for all [x, y] in R, perform powerlaw fit i.e. y = C * x^{−ν}
    return ν
end function
while |G_s| < |G| * r do
    w ← w ∈ V_0
    ν ← 0
    for i = 1 to t_0 do
        w ← MERW(w, ν)
        R ← R ∪ [d_w, (Σ_{v∈N(w)} d_v)/d_w]
        ν ← CalcNu(R)
    end for
    w_i ← w
    for i = 1 to t do
        w ← MERW(w_i, ν)
        V_s ← V_s ∪ {w_i}
        E_s ← E_s ∪ {(w_i, w)}
        R ← R ∪ [d_w, (Σ_{v∈N(w)} d_v)/d_w]
        ν ← CalcNu(R)
        w_i ← w
    end for
end while
```



Algorithm flowchart (left) and estimated correlation coefficient ν as a function of time (right). Note that the estimated ν converges stably to a fixed value with time.

Table of Contents

Experiment Objectives

Q1 : How can we estimate top k Eigenvector centrality scores with limited information?
Q2 : How can we make a GOOD sampling for estimation of global properties of a given graph?

$\Rightarrow$ *Maximal Entropy Random Walk Sampling*

Estimation of Eigenvector Centrality

Top 10 Eigenvector centrality nodes in ca-CondMat

Rank	Node ID	Scores
1	789	0.2254
2	800	0.2174
3	1860	0.1672
4	227	0.1634
5	1369	0.1391
6	3527	0.1384
7	1855	0.1159
8	852	0.1140
9	1858	0.1134
10	1854	0.1064

Estimation of Eigenvector Centrality

Top 10 Eigenvector centrality nodes in ca-CondMat

Rank	Node ID	Scores
1	789	0.2254
2	800	0.2174
3	1860	0.1672
4	227	0.1634
5	1369	0.1391
6	3527	0.1384
7	1855	0.1159
8	852	0.1140
9	1858	0.1134
10	1854	0.1064

Rank	Node ID	Scores
1	16988	0.4378
2	16983	0.3539
3	16986	0.3383
4	16979	0.2474
5	14558	0.2460
6	16989	0.2418
7	16990	0.2404
8	16987	0.2340
9	16980	0.2296
10	16985	0.2215

Rank	Node ID	Scores
1	8121	0.3533
2	15353	0.3533
3	15354	0.3533
4	15351	0.3533
5	15349	0.3533
6	15350	0.3533
7	15352	0.3533
8	15348	0.3533
9	9015	0.0154
10	9013	0.0154

Sampled graph of Metropolis-Hastings RW(left) and Forest Fire(right)

Estimation of Eigenvector Centrality

Top 10 Eigenvector centrality nodes in ca-CondMat

Rank	Node ID	Scores
1	789	0.2254
2	800	0.2174
3	1860	0.1672
4	227	0.1634
5	1369	0.1391
6	3527	0.1384
7	1855	0.1159
8	852	0.1140
9	1858	0.1134
10	1854	0.1064

Rank	Node ID	Scores
1	839	0.3157
2	6703	0.2531
3	1071	0.2523
4	2698	0.2313
5	3598	0.2116
6	1085	0.2028
7	541	0.1837
8	6710	0.1655
9	4955	0.1611
10	242	0.1579

Rank	Node ID	Scores
1	789	0.2573
2	800	0.2474
3	1860	0.1955
4	227	0.1940
5	1369	0.1560
6	3527	0.1526
7	852	0.1366
8	1858	0.1355
9	1855	0.1341
10	1854	0.1290

Sampled graph of General RW(left) and Maximal Entropy RW(right)

Estimation of Eigenvector Centrality

Top 10 Eigenvector centrality nodes in ca-CondMat

Rank	Node ID	Scores
1	789	0.2254
2	800	0.2174
3	1860	0.1672
4	227	0.1634
5	1369	0.1391
6	3527	0.1384
7	1855	0.1159
8	852	0.1140
9	1858	0.1134
10	1854	0.1064

Rank	Node ID	Scores
1	839	0.3157
2	6703	0.2531
3	1071	0.2523
4	2698	0.2313
5	3598	0.2116
6	1085	0.2028
7	541	0.1837
8	6710	0.1655
9	4955	0.1611
10	242	0.1579

Rank	Node ID	Scores
1	789	0.2573
2	800	0.2474
3	1860	0.1955
4	227	0.1940
5	1369	0.1560
6	3527	0.1526
7	852	0.1366
8	1858	0.1355
9	1855	0.1341
10	1854	0.1290

Sampled graph of General RW(left) and Maximal Entropy RW(right)

Estimation of Eigenvector Centrality

Top 10 Eigenvector centrality nodes in ca-CondMat

Rank	Node ID	Scores
1	789	0.2254
2	800	0.2174
3	1860	0.1672
4	227	0.1634
5	1369	0.1391
6	3527	0.1384
7	1855	0.1159
8	852	0.1140
9	1858	0.1134
10	1854	0.1064

Rank	Node ID	Scores
1	839	0.3157
2	6703	0.2531
3	1071	0.2523
4	2698	0.2313
5	3598	0.2116
6	1085	0.2028
7	541	0.1837
8	6710	0.1655
9	4955	0.1611
10	242	0.1579

Rank	Node ID	Scores
1	789	0.2573
2	800	0.2474
3	1860	0.1955
4	227	0.1940
5	1369	0.1560
6	3527	0.1526
7	852	0.1366
8	1858	0.1355
9	1855	0.1341
10	1854	0.1290

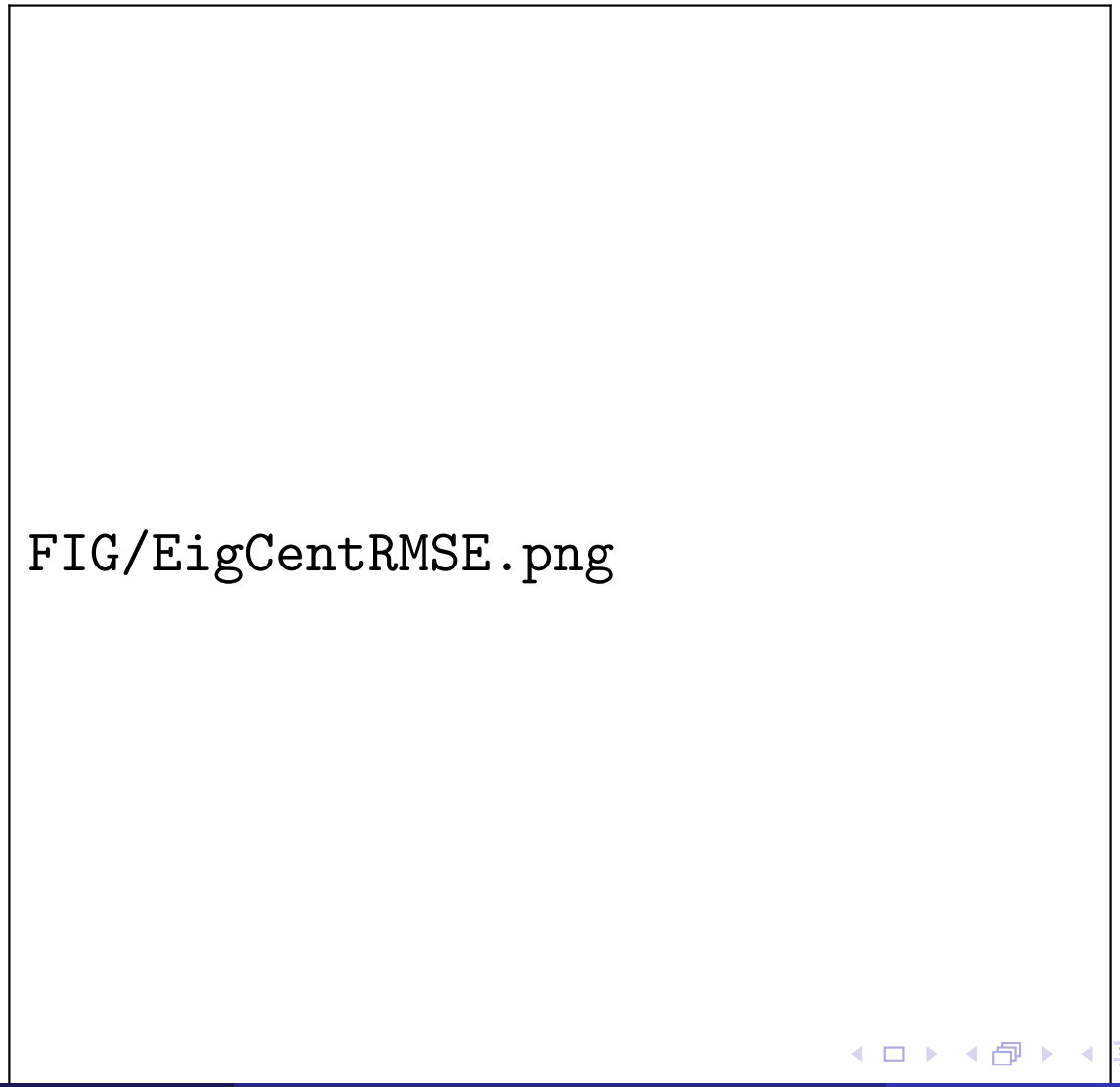
Sampled graph of General RW(left) and Maximal Entropy RW(right)

The order and the scores of nodes of **Maximal Entropy RW** almost match with the original graph!

Estimation of Eigenvector Centrality

Estimation of top 10 Eigenvector Centrality scores

GRW = General RW, MERW = Maximal Entropy RW, MHRW = Metropolis Hastings RW, FF = Forest Fire



Estimation of Global Properties using MERW

1. Evaluation Criteria
   - Degree Distribution
   - Clustering Coefficient
   - Eigenvector Centrality distribution
   - Eigenvalue distribution
   - Hop Plot
2. Evaluation Method:

Kolomogorov-Smirnov D-statistic

$$D = max_x\{|F'(x) - F(x)|\}$$

x : random variable

$F(x)$: (normalized) cumulative distribution function

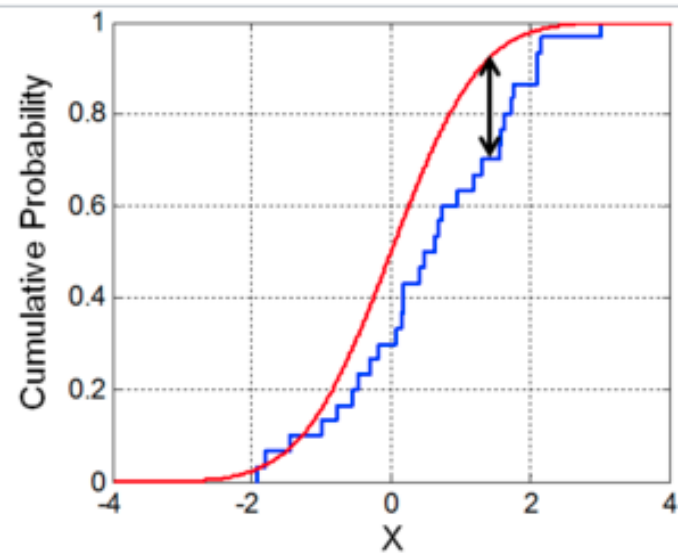
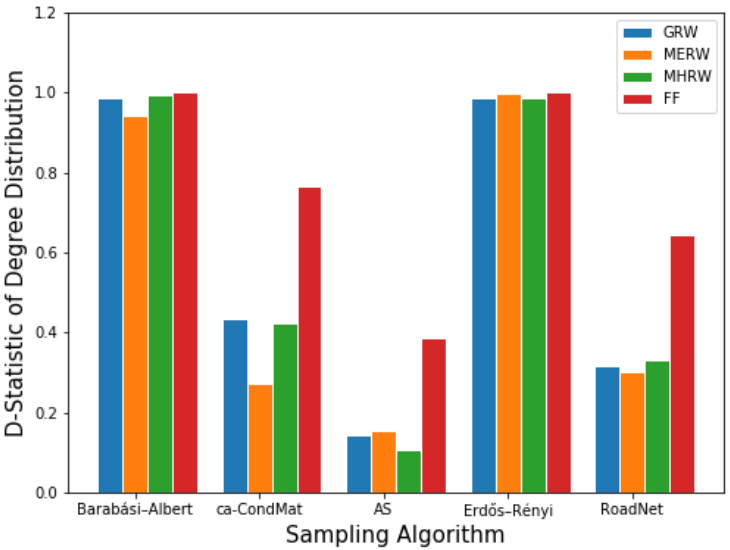


Figure from Wikipedia, black arrow corresponds to the D-statistic. **Lower D indicates better agreement** between two distributions

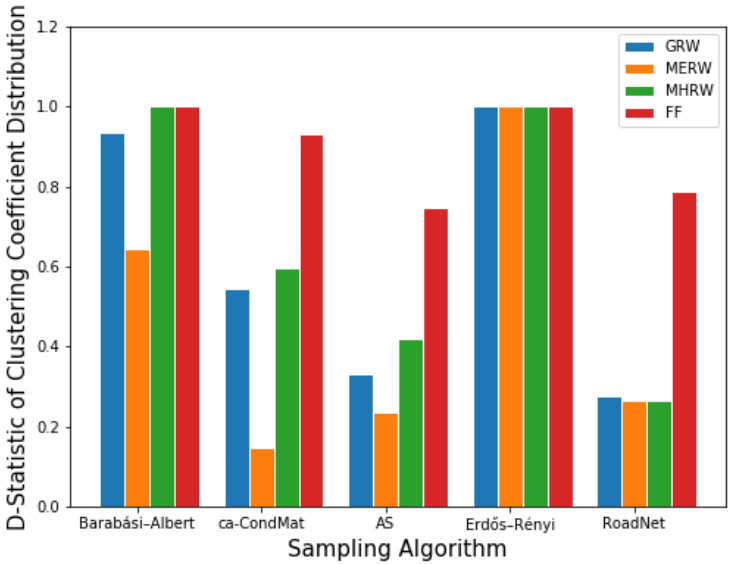
Estimation of Global Properties using MERW

How is the degree centrality distributed?
How skewed is the clustering behavior in the central nodes?

GRW = General RW, MERW = Maximal Entropy RW, MHRW = Metropolis Hastings RW, FF = Forest Fire



Degree Distribution



Clustering Coefficient

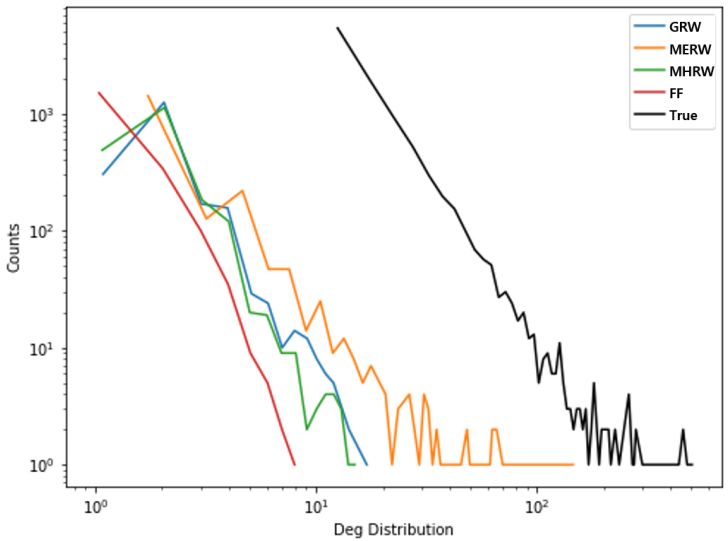
Metropolis Hastings RW and Forest Fire does not capture clustering coefficient and degree distribution effectively
Maximal Entropy RW best estimates Clustering Coefficient Distribution!

Estimation of Global Properties using MERW(Detail)

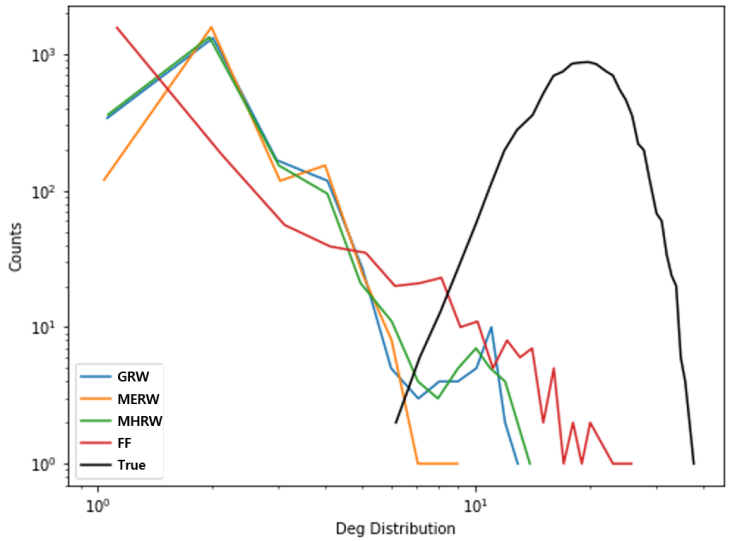
Why does Forest Fire perform poorly in degree estimation?

$\Rightarrow$ Degree Distribution

ES0 = General RW, MERW = Maximal Entropy RW, MHRW = Metropolis Hastings RW, FF = Forest Fire



Barabasi-Albert Model



Erdos-Renyi Model

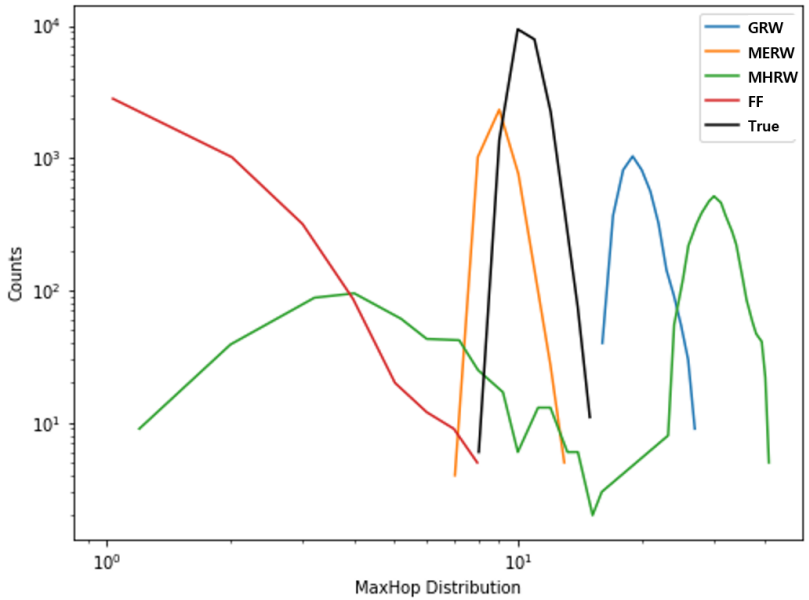
Power-law tail?

Estimation of Global Properties using MERW(Detail)

Why does Metropolis-Hastings perform poorly in clustering coefficient estimation?

⇒Radius plot

ES0 = General RW, MERW = Maximal Entropy RW, MHRW = Metropolis Hastings RW, FF = Forest Fire



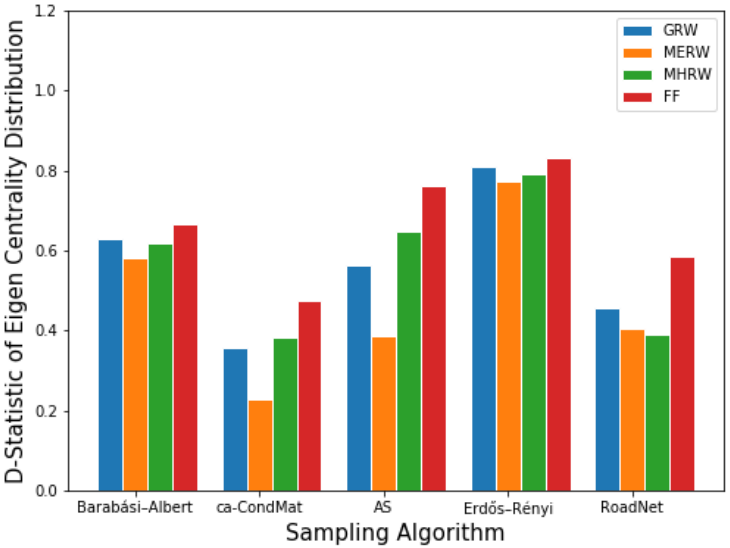
ca-CondMat

Disconnected Componets?

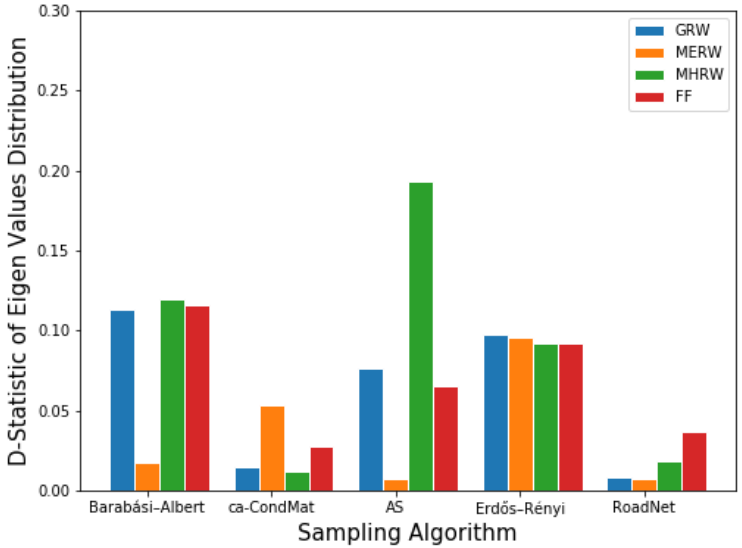
Estimation of Global Properties using MERW

How is the Eigenvector centrality distributed?
How prevalent are the leading Eigenmodes?

GRW = General RW, MERW = Maximal Entropy RW, MHRW = Metropolis Hastings RW, FF = Forest Fire



Eigenvector Centrality Distribution



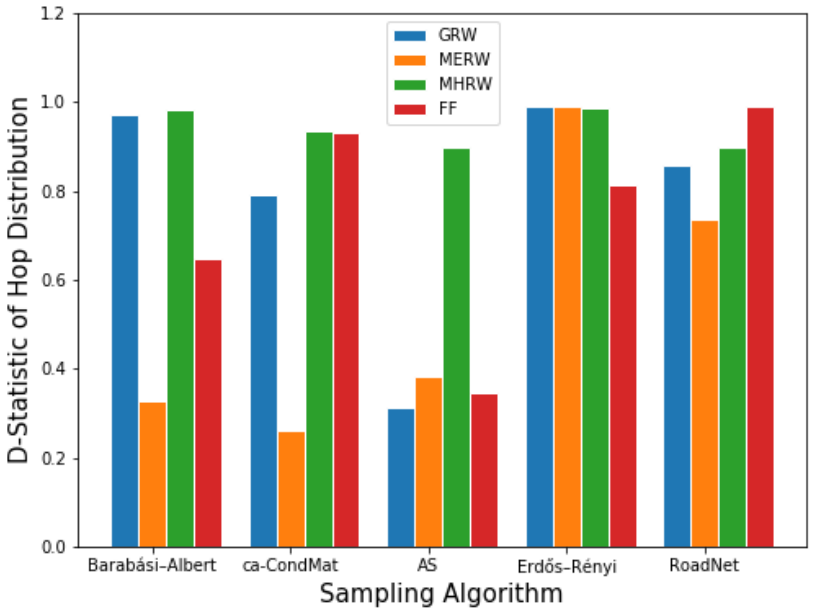
Eigenvalue Distribution

Maximal Entropy RW best estimates Eigenvector Centrality!

Estimation of Global Properties using MERW

How is the radius distributed?

GRW = General RW, MERW = Maximal Entropy RW, MHRW = Metropolis Hastings RW, FF = Forest Fire



Hop Plot

Overall, **Maximal Entropy Random Walk Sampling** gives the best estimation of the given graph

Table of Contents

Conclusion

- Estimation of Eigenvector Centrality is an important task
  - Google PageRank, Katz centrality
  - Biological implications
- **Maximal Entropy Random Walk Sampling(MERW)**
  - novel and powerful tool derived from information theory and physics
  - exact detection of Eigenvector Centralities
  - good estimates of global properties of a given graph
  - highly scalable, since it only needs local information of the graph