

Really simple idea.  
Just generate a graph  
by these Adjacency Matrix  
as a random and probability  
add nodes, edges based on  
params  $(A, B, C, D)$

# R-MAT: A Recursive Model for Graph Mining

Deepayan Chakrabarti\*

Yiping Zhan†

Christos Faloutsos‡

by 1: I wonder if anyone has  
taken that graph and determined these params  
before?  
by 1: I think there's something  
3S. → would be interesting to characterize each  
of our graphs like this...

## Abstract

How does a ‘normal’ computer (or social) network look like? How can we spot ‘abnormal’ sub-networks in the Internet, or web graph? The answer to such questions is vital for outlier detection (terrorist networks, or illegal money-laundering rings), forecasting, and simulations (“how will a computer virus spread?”).

The heart of the problem is finding the properties of real graphs that seem to persist over multiple disciplines. We list such “laws” and, more importantly, we propose a simple, parsimonious model, the “recursive matrix” (R-MAT) model, which can quickly generate realistic graphs, capturing the essence of each graph in only a few parameters. Contrary to existing generators, our model can trivially generate weighted, directed and bipartite graphs; it subsumes the celebrated Erdős-Rényi model as a special case; it can match the power law behaviors, *as well as* the deviations from them (like the “winner does not take it all” model of Pennock et al. [20]). We present results on multiple, large real graphs, where we show that our parameter fitting algorithm (AutoMAT-fast) fits them very well.

## 1 Introduction

Graphs, networks and their surprising regularities/laws have been attracting significant interest recently. The World Wide Web, the Internet topology and Peer-to-Peer networks follows surprising power-laws [5, 10, 3], exhibit strange “bow-tie” or “jellyfish” structures [5, 23], while still having a small diameter [2]. Finding patterns, laws and regularities in large real networks has numerous applications, from criminology and law enforcement [8] to analyzing virus propagation patterns [19] and understanding networks of regulatory genes and interacting proteins [3] and so on.

Discovering and listing such laws is only the first step. Ideally, we would like a generative model with the following properties:

- **Parsimony**: It would have a few only parameters.
- **Realism**: It would only generate graphs that obey the above “laws”, and it would match the properties of real graphs (degree exponents, diameters etc.) with the appropriate values of its parameters.
- **Generation speed**: it would generate the graphs quickly, ideally, linearly on the number of nodes and edges.

This is exactly the main part of this work. We propose the *Recursive Matrix* (R-MAT) model, which naturally generates power-law (or “DGX” [4]) degree distributions. We show that it naturally leads to small-world graphs; it is recursive (=self-similar), and it has only a small number of parameters.

The rest of this paper is organized as follows: Section 2 surveys the existing graph laws and generators. Section 3 presents the idea behind our method and its algorithms. Section 4 gives the experimental results, where we show that R-MAT successfully mimics large real graphs. We conclude in Section 5.

## 2 Background and Related Work

A *graph*  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , is a set  $\mathcal{V}$  of  $N$  nodes, and a set  $\mathcal{E}$  of  $E$  edges between them. The edges may be undirected (like the network of Internet routers and their physical links) or directed (like the network of who-trusts-whom in the *epinions.com* database [21]). *Bipartite graphs* have edges between two sets of nodes, like, for example, the graph of the movie-actor database (*www.imdb.com*).

**Patterns and “Laws”**: Skewed distributions, and power laws of the form  $y = x^a$ , appear very often. Power-laws have been observed for the degree distributions of the Internet, the WWW and the citation graph, the distribution of “bipartite cores” ( $\approx$  communities), the eigenvalues of the adjacency matrix and others [10, 13, 2]. Recently, Pennock et al. [20] observed deviations from power-laws for the Web graph, which are well-modeled by the truncated, discretized lognormal (“DGX”) distribution of Bi et al. [4]. **Graphs also exhibit a strong “community” effect** [11, 14]. Most real graphs have surprisingly small diameters: the “six degrees of separation” for the social network, 19 for the Web, and small values for the Internet AS graph [2, 23]. Apart from these, there are many other measures such as clustering coefficient, expansion, resilience, prestige, influence, stress and so on [7, 12, 22, 18].

Broder et al. [5] show that the WWW has a “bow-tie” structure with 4 roughly equal parts. Tauro et al. [23] find that the Internet topology is organized as a set of concentric circles around a small core, like a “Jellyfish”.

**Graph Generators**: The earliest, and most famous graph generating model is by Erdős and

\*School of Computer Science, CMU

†Dept. of Biological Sciences and School of Computer Science, CMU

‡School of Computer Science, CMU

Rényi. However it provably violates the power laws above. Recent graph generators can be grouped in two classes: *degree based* and *procedural*. Given a degree distribution (typically following a power-law), the degree-based ones try to find a graph that matches it [2, 17], but without giving any insights about the graph or trying to match other criteria (like small diameter, eigenvalues etc.).

On the other hand, procedural generators try to find simple mechanisms to generate graphs that match a property of the real graphs and, typically, the power law degree distribution. This is the class that our proposed R-MAT method belongs to. The typical representative here is the Barabasi-Albert (BA) method with the “*preferential attachment*” idea: keep adding nodes; new nodes prefer to connect to existing nodes with high degrees. However, this only gives power laws of exponent 3. Many modifications and alternatives to the basic idea have been proposed; some generators also include the geometrical layout of nodes in their models [1, 2, 17, 20, 6]. The BRITE graph generator [15] uses components from several of the above models.

In general, all of the above generators fail to meet one or more of the following goals: (a) the generator should be procedural (b) it should be able to generate all types of graphs (directed/undirected, bipartite, weighted) (c) it should match both power-law degree distributions and the “unimodal” distributions observed by Pennock et al. [20] (d) it should satisfy more criteria (like diameter, eigenvalue plots), in addition to the degree distribution.

A related field is that of relational learning [9]; however, this focuses on finding structure at a more local level while our work focuses on the global level. Other topics of interest involving graphs include graph partitioning, frequent subgraph discovery, finding cycles in graphs, and many others. These address interesting problems, and we are investigating their use in our work.

### 3 Proposed Method

Several previous graph generators have been described in Section 2, but they all fail in one aspect or another. The goals a graph generator should achieve are that the generated graph should:

- ( $g1$ ) match the degree distributions (power laws or not)
- ( $g2$ ) exhibit a “community” structure
- ( $g3$ ) have a small diameter, and match other criteria

**Main Idea:** We provide a method which fits both unimodal and power-law graphs using very few parameters. Our method, called **Recursive MATrix**,

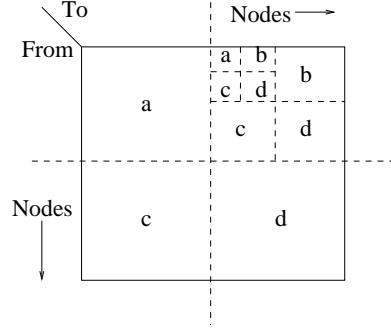


Figure 1: The **R-MAT** model

Symbol	Meaning
$N$	Number of nodes in the real graph
$2^n$	Number of nodes in the R-MAT graph
$E$	Number of edges in the real graph, and in the R-MAT generated graph <i>after</i> duplicate elimination
$(a, b, c, d)$	Probabilities of an edge falling into partitions in the R-MAT model. $a + b + c + d = 1$ .

Table 1: Table of Symbols

or R-MAT for short, generates the graph by operating on its adjacency matrix in a recursive manner.

#### 3.1 Fast Algorithm to generate Directed Graphs:

The adjacency matrix  $\mathbf{A}$  of a graph of  $N$  nodes is an  $N * N$  matrix, with entry  $a(i, j) = 1$  if the edge  $(i, j)$  exists, and 0 otherwise. The basic idea behind **R-MAT** is to recursively subdivide the adjacency matrix into four equal-sized partitions, and distribute edges within these partitions with a unequal probabilities: starting off with an empty adjacency matrix, we “drop” edges into the matrix one at a time. Each edge chooses one of the four partitions with probabilities  $a, b, c, d$  respectively (see Figure 1). Of course,  $a + b + c + d = 1$ . The chosen partition is again subdivided into four smaller partitions, and the procedure is repeated until we reach a simple cell ( $=1 \times 1$  partition). This is the cell of the adjacency matrix occupied by the edge. The number of nodes in the R-MAT graph is set to  $2^n$ ; typically  $n = \lceil \log_2 N \rceil$ . There is a subtle point here: we may have *duplicate* edges (ie., edges which fall into the same cell in the adjacency matrix), but we only keep one of them. To smooth out fluctuations in the degree distributions, we add some noise to the  $(a, b, c, d)$  values at each stage of the recursion and then renormalize (so that  $a + b + c + d = 1$ ). Table 1 shows the symbols used in the paper.

**3.2 Discussion: Meeting the Goals:** Intuitively, our technique is generating “communities” in the graph.

Typically,  $a \geq b$ ,  $a \geq c$ ,  $a \geq d$ .

- The partitions  $a$  and  $d$  represent separate groups of nodes which correspond to communities (say, soccer and automobile enthusiasts).
- The partitions  $b$  and  $c$  are the *cross-links* between these two groups; edges there would denote friends with separate interests.
- The recursive nature of the partitions means that we automatically get sub-communities within existing communities (say, motorcycle and car enthusiasts within the automobile group).

The third bullet results in “communities within communities” (goal  $g2$ ). The skew in the distribution of edges between the partitions ( $a \geq d$ ) leads to lognormals and the DGX distribution (goal  $g1$ ). We shall show experimentally that R-MAT also generates graphs with small diameter and matching other criteria as well (goal  $g3$ ).

**3.3 Parameter fitting with AutoMAT-fast:** The R-MAT model can be considered as a *binomial cascade* in two dimensions. We can calculate the expected number of nodes  $c_k$  with out-degree  $k$ :

$$c_k = \binom{E}{k} \sum_{i=0}^n \binom{n}{i} [p^{n-i}(1-p)^i]^k [1 - p^{n-i}(1-p)^i]^{E-k}$$

where  $2^n$  is the number of nodes in the R-MAT graph and  $p = a + b$ . Fitting this to the observed outdegree distribution gives us the estimated values for  $p = a + b$  (and similarly  $q = a + c$  for the indegree distribution). Conjecturing that the  $a : b$  and  $a : c$  ratios are approximately 75 : 25 (as seen in many real world scenarios), we can calculate the parameters  $(a, b, c, d)$ .

#### 3.4 Extending R-MAT to Undirected Graphs:

An undirected graph must have a symmetric adjacency matrix. We achieve this by generating a directed graph with  $b = c$  and then using a “clip-and-flip” on the resulting adjacency matrix. This involves throwing away the half of matrix above the main diagonal and copying the lower half to it. The effect of this is twofold: first, since  $b = c$ , the number of edges in the final undirected matrix is approximately equal to that in the directed graph; and second, this technique guarantees that the resulting matrix will be symmetric, and hence the corresponding graph will be undirected.

#### 3.5 Extending R-MAT to Bipartite Graphs:

For a bipartite graph, the length and height may be different, and the adjacency matrix will be a rectangle instead of a square. Here too, we set the length and width to be powers of 2, denoted by  $2^{n_1}$  and  $2^{n_2}$ . While dropping edges, we might form a partition with a length(height) of 1; in such a case, further partitions are just top-bottom(left-right) with the appropriate probabilities.

## 4 Experiments

The questions we wish to answer are:

- [Q1] How does R-MAT compare with existing generators for undirected graphs?
- [Q2] How does R-MAT compare with existing generators for directed graphs?
- [Q3] How does R-MAT compare with existing generators for bipartite graphs?

The datasets we use for our experiments are:

*Epinions*: A directed graph of who-trusts-whom from epinions.com [21]:  $N = 75,879$ ;  $E = 508,960$ .

*Epinions-U*: An undirected version of the *Epinions* graph:  $N = 75,879$ ;  $E = 811,602$ .

*Clickstream*: A bipartite graph of Internet users’ browsing behavior [16]. An edge  $(u, p)$  denotes that user  $u$  accessed page  $p$ . It has 23,396 users, 199,308 pages and 952,580 edges.

Apart from degree distributions, we compare the models on their singular value vs. rank plot, first singular vectors (network values) vs. rank plots, “hop-plot” (number of reachable pairs vs. number of hops) and “effective diameter” [18, 23], and stress distribution [12] (the stress of an edge is the number of shortest paths between node pairs that it is a part of). For undirected graphs, eigenvalues and singular values are equivalent; for other graphs, eigenvalues may not exist.

We compared R-MAT to the *AB* [1], *GLP* [6] and *PG* [20] models, chosen for their popularity or recency. All of these are used to generate undirected graphs; they have not been used for directed or bipartite graphs. Thus, we can compare them with R-MAT only on *Epinions-U*. Also, we are unaware of any good parameter-fitting mechanisms for these generators, so for each generator, we exhaustively find the best parameter values. We use *AB+*, *PG+* and *GLP+* to stand for the original algorithms augmented by our parameter fitting.

[Q1] **Undirected Graphs:** We show results in Figure 2 for the *Epinions-U* undirected graph. Notice that R-MAT is very close to *Epinions-U* in all cases, while the competitors are not. Recall that all the y-scales are logarithmic, so small differences actually represent large deviations. The “stress distribution” plot is similar, but is not shown due to lack of space.

[Q2] **Directed Graphs:** We can see from Figure 3 that the match between R-MAT and the *Epinions* dataset is very good. The effective diameter is 4 for both the real graph and for the R-MAT generated graph. The other models considered are not applicable.

[Q3] **Bipartite Graphs:** R-MAT matches the bipartite *Clickstream* dataset very well (Figure 4) including the “un-powerlaw-like” outdegree distribution. The other models are not applicable.

## 5 Conclusions

The goal of this paper was to create a simple, parsimonious graph model to describe real graphs. Our R-MAT model is exactly a step in this direction: we illustrate experimentally that several, diverse real graphs can be well approximated by an R-MAT model with the appropriate choice of parameters. Moreover, we propose a list of natural tests which hold for a variety of real graphs: matching the power-law/DGX distribution for the in- and out-degree; the hop-plot and the diameter of the graph; the singular value distribution; the values of the first singular vector (“Google-score”); and the “stress” distribution over the edges of the graph.

In addition to its realism, our proposed R-MAT model has the following advantages over previous generators:

- It matches real graphs for all the tests mentioned above. This sets it apart from most of the existing graph generators which typically focus on matching only a few of the properties.
- The graphs can be generated very quickly (in  $O(E \log(E) \log(N))$  time)
- We present AutoMAT-fast, a fast algorithm to fit the parameters of this model, so that it can mimic a real graph.
- R-MAT easily subsumes the celebrated Erdős-Rényi model ( $a=b=c=d=0.25$ )
- R-MAT can easily generate realistic weighted graphs (by setting weight=number of duplicate edges), directed graphs and bipartite graphs. None of its competitors can do all the three tasks.
- R-MAT can produce graphs with power-law degree distributions, but it can also produce graphs with degree distributions that match the “winner does not take all” model of [20] - all with just the appropriate choice of parameters.
- R-MAT automatically generates graphs with the “communities within communities” property.

## References

- [1] R. Albert and A.-L. Barabási. Topology of complex networks: local events and universality. *Physical Review Letters*, 85(24), 2000.
- [2] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 2002.
- [3] Albert-László Barabási. *Linked: The New Science of Networks*. Perseus Publishing, first edition, May 2002.
- [4] Z. Bi, C. Faloutsos, and F. Korn. The DGX distribution for mining massive, skewed data. In *KDD*, 2001.
- [5] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web: experiments and models. In *WWW*, 2000.
- [6] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *INFOCOM*, 2002.
- [7] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2002.
- [8] H. Chen, J. Schroeder, R. Hauck, L. Ridgeway, H. Atabaksh, H. Gupta, C. Boarman, K. Rasmussen, and A. Clements. Coplink Connect: Information and knowledge management for law enforcement. *CACM*, 46(1):28–34, January 2003.
- [9] Saso Dzeroski and Nada Lavrac, editors. *Relational Data Mining*. Springer Verlag, 2001.
- [10] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *SIGCOMM*, pages 251–262, 1999.
- [11] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *ACM Conference on Hypertext and Hypermedia*, 1998.
- [12] C. Gkantsidis, M. Mihail, and E. Zegura. Spectral analysis of Internet topologies. In *INFOCOM*, 2003.
- [13] J. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. In *Intl. Conf. on Combinatorics and Computing*, 1999.
- [14] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the web. In *VLDB*, Edinburgh, Scotland, 1999.
- [15] A. Medina, I. Matta, and J. Byers. On the origin of power laws in Internet topologies. In *SIGCOMM*, 2000.
- [16] A. L. Montgomery and C. Faloutsos. Identifying Web browsing trends and patterns. *IEEE Computer*, 34(7), 2001.
- [17] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.
- [18] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. ANF: A fast and scalable tool for data mining in massive graphs. In *SIGKDD*, Edmonton, AB, Canada, 2002.
- [19] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14), 2001.
- [20] David M. Pennock, Gary W. Flake, Steve Lawrence, Eric J. Glover, and C. Lee Giles. Winners don’t take all: Characterizing the competition for links on the Web. *Proceedings of the National Academy of Sciences*, 99(8):5207–5211, 2002.
- [21] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *SIGKDD*, pages 61–70, Edmonton, Canada, 2002.
- [22] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topologies, power laws, and hierarchy. Technical Report 01-746, U. Southern California, 2001.
- [23] S. L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the Internet topology. In *Global Internet, San Antonio, Texas*, 2001.

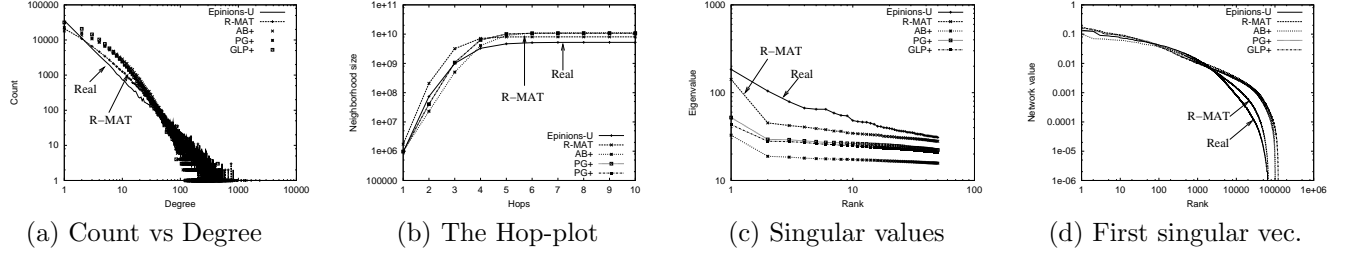


Figure 2: *The Epinions-U Undirected Graph*: The R-MAT plots gives the best fit to the *Epinions-U* graph (solid line) among all the generators.

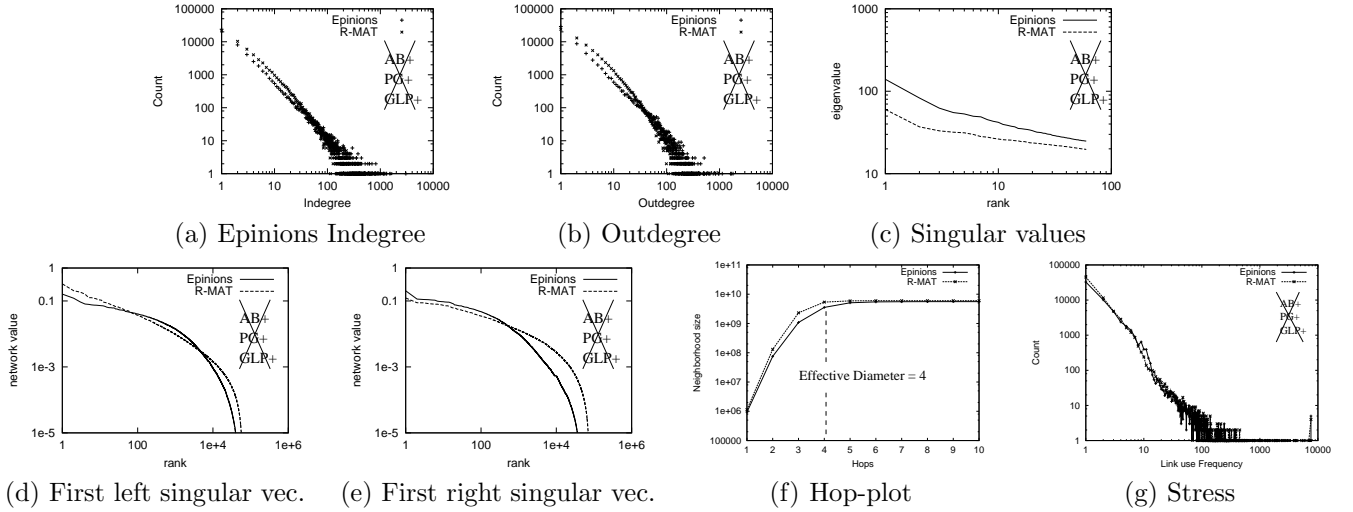


Figure 3: *The Epinions Directed Graph*: The *AB+*, *PG+* and *GLP+* methods **do not apply**. The crosses and dashed lines represent the R-MAT generated graphs, while the pluses and strong lines represent the real graph.

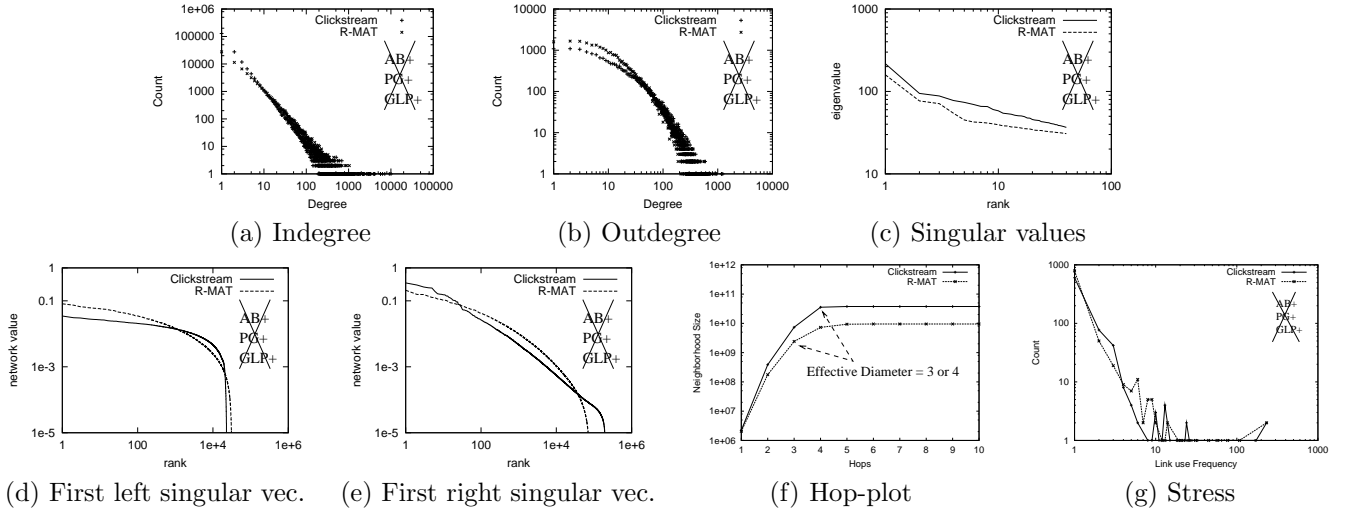


Figure 4: *The Clickstream Bipartite Graph*: The *AB+*, *PG+* and *GLP+* methods **do not apply**. The crosses and dashed lines represent the R-MAT generated graphs, while the pluses and strong lines represent the real graph.