

RTM: Laws and a Recursive Generator for Weighted Time-Evolving Graphs

Leman Akoglu Mary McGlohon Christos Faloutsos
Carnegie Mellon University, School of Computer Science
{lakoglu, mmcgloho, christos}@cs.cmu.edu

Abstract

How do real, weighted graphs change over time? What patterns, if any, do they obey? Earlier studies focus on unweighted graphs, and, with few exceptions, they focus on static snapshots. Here, we report patterns we discover on several real, weighted, time-evolving graphs. The reported patterns can help in detecting anomalies in natural graphs, in making link prediction and in providing more criteria for evaluation of synthetic graph generators. We further propose an intuitive and easy way to construct weighted, time-evolving graphs. In fact, we prove that our generator will produce graphs which obey many patterns and laws observed to date. We also provide empirical evidence to support our claims.

1 Introduction

Static, unweighted graphs have attracted a lot of interest recently, with several fascinating discoveries. Numerous generators also try to mimic these patterns. Very recently, time-evolving graphs have attracted some attention, however, graphs that are both weighted and dynamic have been relatively unexplored. Here we focus on weighted graphs - both static snapshots of them, as well as dynamic properties.

Given a set of edge weights on a graph, a few questions come to mind. What patterns do the weights obey? Do they follow a Gaussian distribution, for a given snapshot in time? How, if at all, is the edge weight related to the popularity of its adjacent nodes? Which of these static patterns persist over time?

In this work, we answer all these questions, and we show that there are some unexpected patterns. In summary, the contributions are the following: (1) We present several new patterns for weighted, time-evolving graphs. (2) We give a simple generative model (*RTM*, for *Recursive Tensor Multiplication*) that generates weighted, time-evolving graphs that obey all the old and new properties. (3) We prove [1] that *RTM* produces several desired characteristics.

The most striking patterns we discover here are: (a) the

first eigenvalue $\lambda_{1,w}(t)$ of the *weighted* adjacency matrix at time t , follows a power law with respect to the total number of edges $E(t)$ at time t , for several time-ticks. (b) a similar power law holds for the first eigenvalue $\lambda_1(t)$ of the 0-1 adjacency matrix, with different slope, of course. (c) for a given time snapshot, the weight $w_{i,j}$ for edge (i, j) is closely related to the total weights w_i and w_j of its adjacent nodes i and j .

The rest of the paper is organized as follows: Section 2 surveys earlier work. Section 3 provides dataset descriptions. Section 4 lists laws and observations. Section 5 gives details of our *RTM* generator. Section 6 presents experimental results. We conclude in Section 7.

2 Related Work

Real-world graph properties: Many impressive patterns including power laws that real-world graphs obey have been discovered in [12, 3, 8, 10, 17]. Studies on spectral properties of power-law graphs can be found in [9, 13, 16, 18].

Graph generators: Several graph generators have been proposed in [7, 2, 19, 12, 11]. See [5] for a detailed survey on graph generators.

3 Data Description

We studied several large real-world weighted graphs described in detail in Table 1. In particular, *BlogNet* contains blog-to-blog links, *NetworkTraffic* records IP-source/IP-destination pairs, along with the number of packets sent. Bipartite networks *Auth-Conf*, *Keyw-Conf*, and *Auth-Keyw* are all from DBLP and have the submission records of authors to conferences with specified keywords. *CampaignOrg* is from the US FEC, a public record of donations between political candidates and organizations.

For *NetworkTraffic* and *CampaignOrg* datasets, the weights on the edges are actual weights representing number of packets and donation amounts, respectively. For the remaining datasets, the edge weights are simply the number of occurrences of the edges. For instance, if author i submits a paper to conference j for the first time, the weight of $e_{i,j}$

Name	N, E, time	Description
<i>BlogNet</i>	60K, 125K, 80 days	Social network of blogs based on citations
<i>NetworkTraffic</i>	21K, 2M, 52 mo.	Network traffic: packets sent from IP source to IP destination
<i>AuthorConference</i>	17K, 22K, 25 yr.	DBLP Author-to-Conference associations
<i>KeywordConference</i>	10K, 23K, 25 yr.	DBLP Keyword-to-Conference associations
<i>AuthorKeyword</i>	27K, 189K, 25 yr.	DBLP Author-to-Keyword associations
<i>CampaignOrg</i>	23K, 877K, 28 yr.	U.S. electoral campaign donations (available from FEC)

Table 1. Weighted datasets studied in this work.

Symbol	Description
\mathcal{G}	Graph representation of datasets
\mathcal{V}	Set of nodes for graph \mathcal{G}
\mathcal{E}	Set of edges for graph \mathcal{G}
N	Number of nodes, or $ \mathcal{V} $
E	Number of edges, or $ \mathcal{E} $
$e_{i,j}$	Edge between node i and node j
$w_{i,j}$	Weight on edge $e_{i,j}$
w_i	Weight of node i (sum of weights of incident edges)
\mathbf{A}	0-1 Adjacency matrix of the un-weighted graph
\mathbf{A}_w	Real-value adjacency matrix of the weighted graph
$a_{i,j}$	Entry in matrix \mathbf{A}
λ_1	Principal eigenvalue of unweighted graph
$\lambda_{1,w}$	Principal eigenvalue of weighted graph
$\mathcal{A}, \mathcal{B}, \mathcal{C}$	Tensors used to illustrate recursive tensor product
$a_{i,j,k}$	Entry of a tensor
\mathcal{I}	Initial tensor in <i>RTM</i> model
\mathcal{G}_A	t-graph (time-evolving graph) represented by tensor \mathcal{A}
\mathbf{D}_t	t^{th} slice of final tensor \mathcal{D} in <i>RTM</i>
s_t	Total weight of \mathbf{D}_t
e_t	Number of edges of \mathbf{D}_t
$W_{\mathcal{D}}$	Total weight of a tensor \mathcal{D} , or $\sum_t s_t$
$\mathbf{s}_{\mathcal{D},r}$	Temporal profile of \mathcal{D} at resolution r
$\mathbf{p}_{\mathcal{D},r}$	Normalized temporal profile of \mathcal{D} at resolution r

Table 2. Table of symbols used in notation.

is set to 1. If the same author later submits another paper to the same conference, the edge weight becomes 2.

Throughout this paper we will use the graph representation of the datasets we study. A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes \mathcal{V} , connected by a set of undirected, weighted edges \mathcal{E} . No multiple edges between two nodes are allowed - however, we will account for repeated edges in edge weights. By nature, none of the datasets we study contain self-loops.

In a weighted graph \mathcal{G} , let $e_{i,j}$ be the edge between node i and node j . We shall refer to these two nodes as the ‘neighboring nodes’ or ‘incident nodes’ of edge $e_{i,j}$. Let $w_{i,j}$ be the weight on edge $e_{i,j}$. The *total weight* w_i of node i is defined as the sum of weights of all its incident edges, that is $w_i = \sum_{k=1}^{d_i} w_{i,k}$, where d_i denotes its degree. As we show later, there is a relation between a given edge weight $w_{i,j}$ and the weights of its neighboring nodes w_i and w_j .

A complete list of the symbols used throughout text is listed in Table 2.

4 Laws and Observations

4.1 LPL: Principal eigenvalue over time

Plotting the largest(principal) eigenvalue of the 0-1 adjacency matrix \mathbf{A} of our datasets over time, we notice that the principal eigenvalue grows following a power law with increasing number of edges. This observation is true especially after the *gelling point*. The ‘gelling point’ is defined to be the point at which a giant connected component (GCC) appears in real-world graphs - after this point, properties such as densification and shrinking diameter become increasingly evident. See [12] for details.

Observation 1 (λ_1 Power Law (LPL)) *In real graphs, the principal eigenvalue $\lambda_1(t)$ and the number of edges $E(t)$ over time follow a power law with exponent less than 0.5, especially after the ‘gelling point’. That is,*

$$\lambda_1(t) \propto E(t)^\alpha, \alpha \leq 0.5$$

We report the power law exponents in Fig. 1. Note that we fit the given lines *after* the gelling point which is shown by a vertical line for each dataset. Notice that the given slopes are less than 0.5, with the exception of the *CampaignOrg* dataset, with slope ≈ 0.53 . This result is in agreement with graph theory. See [1] for details.

4.2 LWPL: Weighted principal eigenvalue over time

Given that unweighted (0-1) graphs follow the λ_1 Power Law, one may ask if there is a corresponding law for weighted graphs. To this end, we also compute the largest eigenvalue $\lambda_{1,w}$ of the *weighted* adjacency matrix \mathbf{A}_w . The entries $w_{i,j}$ of \mathbf{A}_w now represent the actual edge weight between node i and j . We notice that $\lambda_{1,w}$ increases with increasing number of edges following a power law with a higher exponent than that of its λ_1 Power Law. We show the experimental results in Fig. 2.

Observation 2 ($\lambda_{1,w}$ Power Law (LWPL)) *Weighted real graphs exhibit a power law for the largest eigenvalue of the*

weighted adjacency matrix $\lambda_{1,w}(t)$ and the number of edges $E(t)$ over time. That is,

$$\lambda_{1,w}(t) \propto E(t)^\beta$$

In our experiments, the exponent β ranged from 0.5 to 1.6.

4.3 EWPL: Edge Weights Power Law

We observe that the weight of a given edge and weights of its neighboring two nodes are correlated. Our observation is similar to Newton’s Gravitational Law stating that the gravitational force between two point masses is proportional to the product of the masses.

Observation 3 (Edge Weights Power Law(EWPL))

Given a real-world graph \mathcal{G} , ‘communication’ defined as the weight of the link between two given nodes has a power law relation with the weights of the nodes. In particular, given an edge $e_{i,j}$ with weight $w_{i,j}$ and its two neighbor nodes i and j with weights w_i and w_j , respectively,

$$w_{i,j} \propto \left(\sqrt{(w_i - w_{i,j}) * (w_j - w_{i,j})} \right)^\gamma$$

We report corresponding experimental findings in Fig. 3.

5 Generative Recursive Tensor Model (RTM)

How could we have a simple generative model that will obey all the patterns we know so far, as well as the newly discovered ones for weighted graphs? Specifically, we would like the model to exhibit:

1. SUGP: Static Unweighted Graph Properties: (a) small diameter [19], (b) power-law degree distribution [4, 8].
2. SWGP: Static Weighted Graph Properties: (a) the Edge Weight Power Law (EWPL) (Observation 3), (b) the Snapshot Power Law (SPL) [14].
3. DUGP: Dynamic Unweighted Graph Properties: (a) the Densification Power Law (DPL) [12], (b) shrinking diameter [12], (c) the λ_1 Power Law (LPL)(Observation 1), (d) bursty edge additions [15].
4. DWGP: Dynamic Weighted Graph Properties: (a) the $\lambda_{1,w}$ Power Law (LWPL)(Observation 2), (b) bursty weight additions [14], (c) the Weight Power Law (WPL) [14].

At the high level, our idea is to use recursion, in conjunction with tensors (n -dimensional extension of matrices). Recursion and self-similarity naturally lead to modular network behavior (“communities-within-communities”), power laws and bursty traffic. Earlier

work used self-similarity to generate static snapshots of un-weighted graphs [6].

Here, we show how to build a generator that will match all of the properties listed. The idea is to use recursion not only on the adjacency matrix, but also on the *time* dimension. Specifically, we start with a small tensor \mathcal{I} that has 3 sides (‘modes’): (a) senders (b) recipients and (c) time. We call the graph represented by a tensor a ‘t-graph’ that evolves over time (See Fig. 4(a-b)). Then, we recursively substitute every cell (i, j, t) of the original tensor \mathcal{I} , with a copy of itself, and multiply it with the value $a_{i,j,t}$ (See Fig. 4(c) for illustration and Definition 1 for full details). Thanks to the self-similarity of the construct, we expect the resulting tensor to have all the properties that we want.

5.1 Description

For the construction, we choose an initial $(N \times N \times \tau)$ tensor \mathcal{I} with nonzero cells (i, j, t) indicating an edge from node i to node j at time tick t . We initialize the cells so that the initial t-graph(t- for time-evolving) $\mathcal{G}_{\mathcal{I}}$ represented by \mathcal{I} looks like a miniature real-world graph. See [1] for details about initialization.

We propose to use *Recursive Tensor Multiplication* to produce a time-evolving graph. Our method extends Kronecker product¹ of two matrices by adding a third ‘mode’. Kronecker product of two matrices is defined as follows: Given two matrices \mathbf{A} and \mathbf{B} of sizes $(N \times M)$ and $(N' \times M')$, respectively, the Kronecker product of \mathbf{A} and \mathbf{B} , namely matrix \mathbf{C} of dimension $(N * N') \times (M * M')$ is given by

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \dots & a_{1,M}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \dots & a_{2,M}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1}\mathbf{B} & a_{N,2}\mathbf{B} & \dots & a_{N,M}\mathbf{B} \end{pmatrix}$$

Definition 1 (Recursive Tensor Multiplication (RTM))

Given two tensors \mathcal{A} of size $(N \times M \times \tau)$ and \mathcal{B} of size $(N' \times M' \times \tau')$, *Recursive Tensor Multiplication* \mathcal{C} of \mathcal{A} and \mathcal{B} is obtained by replacing each cell $a_{i,j,t}$ of tensor \mathcal{A} with $a_{i,j,t} * \mathcal{B}$. The resulting tensor \mathcal{C} is of size $(N * N') \times (M * M') \times (\tau * \tau')$ such that

$$c_{((i-1)*N+i'),((j-1)*M+j'),((k-1)*\tau+k')} = a_{i,j,k} * b_{i',j',k'}.$$

An example of the Recursive Tensor Multiplication of a $(3 \times 3 \times 3)$ tensor by itself is given in Fig. 4(c).

To generate a growing graph over time, we get the ‘Recursive Tensor Multiplication’ of the initial $(N \times N \times \tau)$

¹Unfortunately, Kronecker product \mathbf{C} of two matrices \mathbf{A} and \mathbf{B} is also called Kronecker Tensor multiplication, despite $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are matrices. To disambiguate, we use the name *RTM* where $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are in fact tensors.

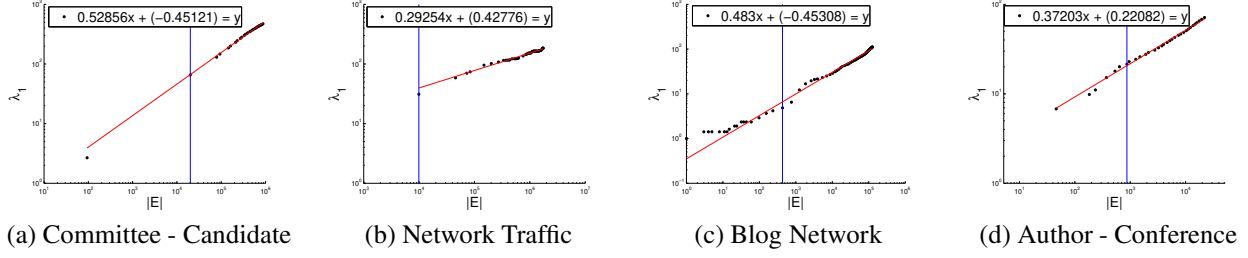


Figure 1. Illustration of the LPL. 1^{st} eigenvalue $\lambda_1(t)$ of the $0-1$ adjacency matrix A versus number of edges $E(t)$ over time. The vertical lines indicate the gelling point.

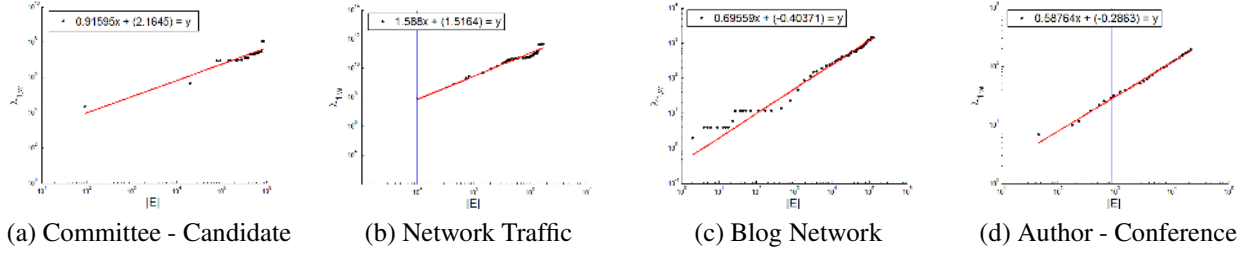


Figure 2. Illustration of the LWPL. 1^{st} eigenvalue $\lambda_{1,w}(t)$ of the *weighted* adjacency matrix A_w versus number of edges $E(t)$ over time. The vertical lines indicate the gelling point.

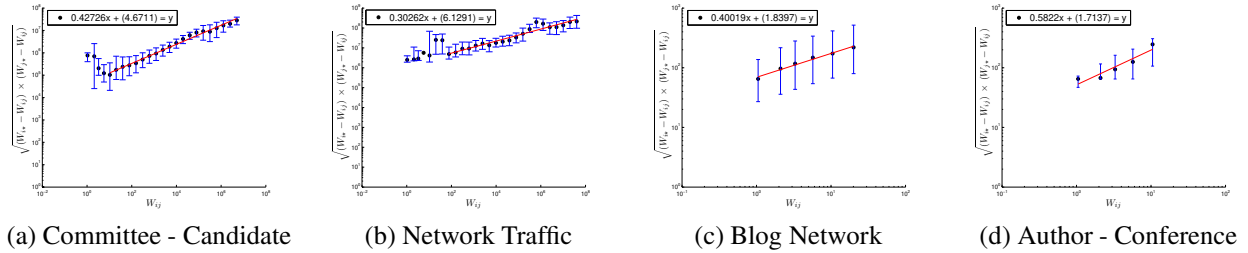


Figure 3. Illustration of the EWPL. Given the weight of a particular edge in the final snapshot of real graphs (x-axis), the multiplication of total weights(y-axis) of the edges incident to two neighboring nodes follow a power law. A line can be fit to the median values after logarithmic binning on the x-axis. Upper and lower bars indicate 75% and 25% of the data, respectively.

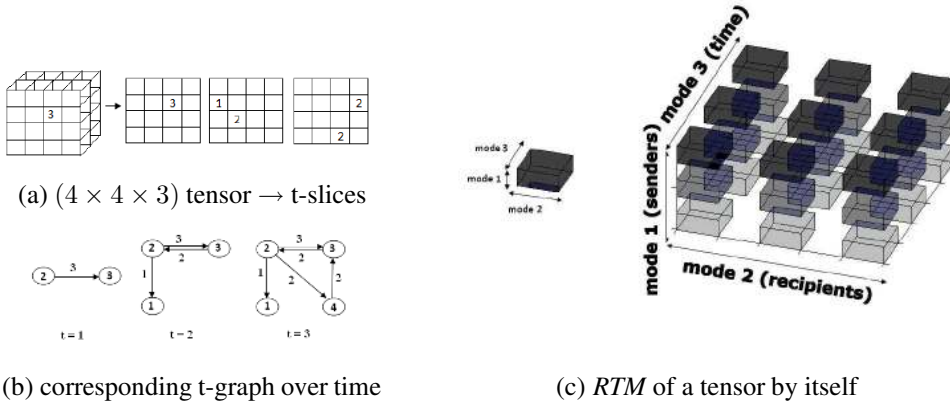


Figure 4. (a) An example for the initial tensor \mathcal{I} of size $(4 \times 4 \times 3)$. The ‘t-slices’ represent the changes on the adjacency matrix at every other time step. (b) The corresponding graph represented by the tensor in part (a). It changes according to the ‘t-slices’ over time. (c) An example $(3 \times 3 \times 3)$ tensor \mathcal{I} is given on the left. Recursive tensor product of \mathcal{I} by itself, that is, the resulting $(3^2 \times 3^2 \times 3^2)$ tensor $\mathcal{D} = \mathcal{I} \textcircled{+} \mathcal{I}$ is given on the right.

tensor \mathcal{I} by itself k times as:

$$\mathcal{I}^k = \mathcal{D} = \underbrace{\mathcal{I} \textcircled{+} \mathcal{I} \textcircled{+} \dots \textcircled{+} \mathcal{I}}_{k \text{ times}}$$

and then we take the final tensor \mathcal{D} to represent our data. The data spans τ^k number of time ticks with N^k nodes. At every time step t ($t = \{1, 2, \dots, \tau^k\}$), we get the t-slice (See Definition 2 below) \mathbf{D}_t of \mathcal{D} , and for each nonzero cell $a_{i,j}$ of \mathbf{D}_t , we add an edge between node i and node j with weight $a_{i,j}$. If the edge already exists, we increase the weight $w_{i,j}$ by the same amount.

Definition 2 (t-slice of a tensor \mathcal{T}) Given a tensor \mathcal{T} of size $(N \times M \times \tau)$, t-slice of \mathcal{T} is a matrix \mathbf{T}_t such that

$$T_t \equiv \mathcal{T}(i, j, t), \quad \forall i, \forall j, \quad 1 \leq i \leq N, 1 \leq j \leq M$$

Definition 3 ((Normalized) temporal (t-) profile of \mathcal{T})

Given a tensor \mathcal{T} of size $(N \times M \times \tau)$, let s_t denote the total weight of its t-slice. Then, the t-profile of \mathcal{T} is a $(1 \times \tau)$ vector, such that $\mathbf{s}_{\mathcal{T},0} \equiv (s_1, s_2, \dots, s_\tau)$. Total weight $W_{\mathcal{T}}$ of \mathcal{T} can be written as $\sum_{t=1}^{\tau} s_t$. Then, normalized t-profile of \mathcal{T} is a $(1 \times \tau)$ vector, such that $\mathbf{p}_{\mathcal{T},0} \equiv (\frac{s_1}{W_{\mathcal{T}}}, \frac{s_2}{W_{\mathcal{T}}}, \dots, \frac{s_\tau}{W_{\mathcal{T}}})$.

5.2 Theorems and Proofs

Theorem 1 (Self-similar and Bursty Edge/Weight Additions)

Let edge/weight additions for \mathcal{I} with $\mathbf{p}_{\mathcal{I},0}$ be self-similar and bursty for which the slope of the entropy plot is

$$\text{slope} = H(\mathbf{p}_{\mathcal{I},0}) = - \sum_{i=1}^{\tau} \mathbf{p}_{\mathcal{I},0}(i) \log_2(\mathbf{p}_{\mathcal{I},0}(i)),$$

After k iterations of RTM, edge/weight arrivals over time for \mathcal{D} are also self-similar and bursty. The slope of the entropy plot over all aggregation levels r of \mathcal{D} is equal to

$$\text{slope} = H(\mathbf{p}_{\mathcal{D},r}) = H(\mathbf{p}_{\mathcal{I},0}), \quad \forall r$$

where $H(\mathbf{p}_{\mathcal{D},r})$ is the slope of the entropy plot at aggregation level r . Furthermore, the slope does not change with the value of k , that is, burstiness is independent of scale.

Proof See [1]. ■

Theorem 2 (Weight Power Law (WPL)) If the initial graph $\mathcal{G}_{\mathcal{I}}$ exhibits the WPL [14] at all time ticks, that is, number of edges $E(t)$ and total weight $W(t)$ over time follow a power law with exponent α , $\mathcal{G}_{\mathcal{D}}$ shows the same property at time ticks $1, \tau^1, \tau^2, \dots, \tau^k$ with exactly the same exponent α .

Proof See [1]. ■

6 Experimental Results

As a comparison with real-world data, we give the plots showing reported laws for *BlogNet* and the plots our model generated for $N = 10$, $\tau = 2$ and $k = 3$ in Fig.5. In particular, we show (a) the Densification Power Law (DPL); (b) the Weight Power Law (WPL); (c) the λ_1 Power Law (LPL), (d) the $\lambda_{1,w}$ Power Law (LWPL) and finally, (e) the Edge Weight Power Law (EWPL) from left to right. Note that characteristics matched by RTM include both those from previous work as well as additional patterns discovered in this work. Other desired characteristics such as small and shrinking diameter, the gelling point, etc. are also matched, but omitted here for brevity.

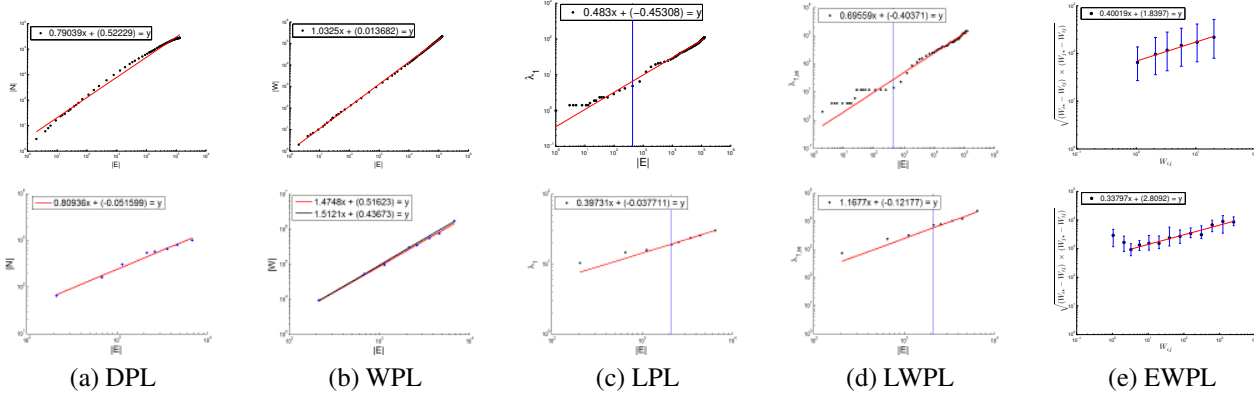


Figure 5. Plots showing related laws that real-world graphs obey for *BlogNet* on the 1st row and for our *RTM* generator on the 2nd row.

7 Conclusion

This is one of the few papers that focus on real, weighted, time-evolving graphs. The contributions are the following:

1. We discovered several patterns that such graphs follow, like the *LPL*, and the *EWPL*.
2. We gave a simple, recursive generator that mimicks a long list of laws observed on weighted time-evolving graphs, as well as on unweighted and/or static graphs.
3. We rigorously proved [1] that *RTM* produces many desired characteristics.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. IIS-0705359 CNS-0721736. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

References

- [1] L. Akoglu, M. McGlohon, and C. Faloutsos. Rtm: Laws and a recursive generator for weighted time-evolving graphs. *Carnegie Mellon University Technical Report*, Oct, 2008.
- [2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *CoRR*, cond-mat/0106096, 2001.
- [3] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the world wide web. *Nature*, 401:130–131, 1999.
- [4] A.-L. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005.
- [5] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1), 2006.
- [6] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. *SIAM Int. Conf. on Data Mining*, Apr. 2004.
- [7] P. Erdos and A. Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hungary. Acad. Sci.*, 5:17–61, 1960.
- [8] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *SIGCOMM*, pages 251–262, Aug-Sept. 1999.
- [9] I. J. Farkas, I. Derényi, A.-L. Barabási, and T. Vicsek. Spectra of ‘real-world’ graphs: Beyond the semi-circle law. *Physical Review E*, 64, 2001.
- [10] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The Web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–17, 1999.
- [11] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. *PKDD*, pages 133–145, 2005.
- [12] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. of ACM SIGKDD*, pages 177–187, Chicago, Illinois, USA, 2005. ACM Press.
- [13] F. C. L. Lu and V. Vu. Eigenvalues of random power law graphs. *Annals of Combinatorics*, 33:7:21, 2003.
- [14] M. McGlohon, L. Akoglu, and C. Faloutsos. Weighted graphs and disconnected components: Patterns and a generator. In *ACM SIGKDD*, Las Vegas, Aug 2008.
- [15] M. McGlohon, J. Leskovec, C. Faloutsos, N. Glance, and M. Hurst. Finding patterns in blog shapes and blog evolution. In *International Conference on Weblogs and Social Media*, Boulder, Colo., Mar. 2007.
- [16] M. Mihail and C. Papadimitriou. The eigenvalue power law, 2002.
- [17] M. E. J. Newman. Power laws, pareto distributions and zipf’s law, December 2004.
- [18] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. Powerlaws and the as-level internet topology, 2003.
- [19] D. J. Watts and S. H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998.