Rumor Spreading in LiveJournal

Saket Sathe

Mini-Project Report, Dynamical Networks

July 4, 2008

Abstract

Online social networks have gained importance in recent years. Furthermore, there is a need for designing smart applications for social networks which take into account the behaviour of dynamical processes over these networks. This requires structural knowledge of the network helpful in understanding the dynamical process. Here we study a broad category of such processes called *rumor spreading* processes. We simulate a typical rumor spreading scenario on a real social network graph of 5.2 million nodes and 72 million edges. We compare the results of this simulation to two synthetically generated Erdős-Rényi [1] and power-law random graphs. Our simulation shows that the behavior of the rumor spreading process is considerably different in social networks as compared to the one observed on above mentioned synthetic random graphs. These simulations have possible implications to applications in viral advertising, social marketing, worm attacks, online political campaigns, peer-to-peer communication networks.

1 Introduction

In recent years online social networks have started becoming extremely popular. Now there are dedicated websites like Orkut [2], Facebook [3], hi5 [4], LiveJournal [5], etc. which provide online social networking capabilities. For this reason the study of online social networks have gained importance. It also helps in designing and deploying applications that are capable of exploiting the niceties of these social networks. Previous studies of social networks have burgeoned trends in user behavior. For example, adjacent users in a social network tend to trust each other more than anyone else. Furthermore, users with a high number of friends tend to connect with other users with high number of friends. These trends in the network structure are different as compared to the Web [6]. Although social networks obey the power-law property they also obey, unlike the Web, the small-world property [7]. Networks bearing this property have all the users (vertices) in the network connected to each other with a small distance (counted as the number of edges between them) with high probability.

As users tend to trust their neighbours they become more susceptible to be victims of network flood attacks or worm attacks in social networks. These worms generally trick the user into doing something that causes the worm to spread to the users friends or connections [8]. Furthermore, there are legitimate cases where users pass on messages to their friends as a part of viral marketing or political campaigning for elections. Also, there are websites that pay a user to promote marketing of their products [9]. All the situations described above can be considered under the broad category of rumor spreading. Rumor spreading is a phenomenon which occurs in networks, like the social network, and results in spreading of information or misinformation based on actions by individual agents in the network. More scientifically, rumor spreading is studied as a dynamical process occurring on top of a network or social network. Obviously, the rate and the penetration of rumor spreading depends on the structure of the underlying network [10].

Largely the dynamics of rumor spreading is studied mainly by physicists [11, 12, 13, 14]. They model the rumor by differential equations (mean-field rate equations) and solve them numerically to obtain the final steady state of the model [10]. In addition to this, they give simulation results on small synthetically generated graphs. In a rumor spreading model, generally the vertices in a network could be in any of the predefined possible states. In each case the state of a vertex depends on states of vertices connected to it. There is also a *final* state, if a vertex reaches that state it never changes its state ever after, thus allowing the rumor to die out. A detailed explanation of this process will be given in Section (2.2).

Admittedly, to the best of our knowledge there is no practical evidence provided by simulation of such rumors on large graphs obtained from crawls of online social networks. Here we study the dynamics of rumor spreading on a social blogging platform called LiveJournal [5]. We simulate the rumor spreading process on the 5.2 million vertex and 72 million edge graph of LiveJournal obtained from the crawl performed by [15]. We also compare the results to a rumor spreading process simulated on synthetically generated Erdős-Rényi [1] and power-law random graphs. We find that the small world property effects the spreading of rumor considerably, and the number of effected users in a small world network are on an average more than the other networks.

The rest of this report is organized as follows. In Section (2.1) we analyse the graph structure of LiveJournal. In Section (2.2) we give details about the rumor spreading model on homogeneous networks. Then in Section (3) we provide an extensive description of the experimental results followed by the discussion of the results and implementation details. Finally, we conclude the report in Section (4).

2 LiveJournal – Social Blogging Platform

LiveJournal [5] is a social blogging platform. It enables users to create a blog and, along with the blog also allows users to manage their own social network. LiveJournal's homepage is shown in Figure (2). Every author/user in LiveJournal can have his/her profile. Generally an author's profile can be accessed as `http://[username].livejournal.com/profile`. A typical member profile is shown in Figure (2). As it can be seen the profile clearly shows friends of the current member and also provides links to them. Furthermore, friendships in LiveJournal can be considered as *directed links* since LiveJournal allows users to link to each other without their consent. Thus the graph produced by LiveJournal is directed, unlike graphs that are produced by other popular social networks like Orkut [2] or FaceBoook [3]. LiveJournal also offers an API for programatically viewing an author's profile Mislove *et al.* [15] have used this API to crawl LiveJournal's social graph. We have performed all the experiments on the graph provided by this crawl in December 2006. The crawl consists of 5,284,457 (over 5 million) users and 77,402,652 (over 77 million) number of friend links.

2.1 Analysis of the Graph Structure

In this section we describe some important properties of graphs in general and the LiveJournal graph in particular. First we define some important properties that are of interest to us and then we give their values for the LiveJournal graph along with explanation.

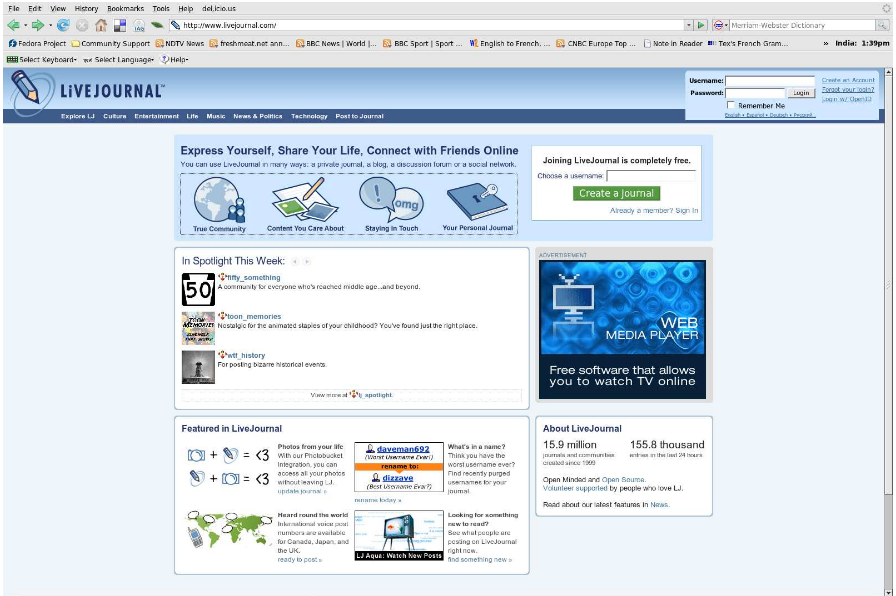


Figure 1: Homepage of LiveJournal.com.

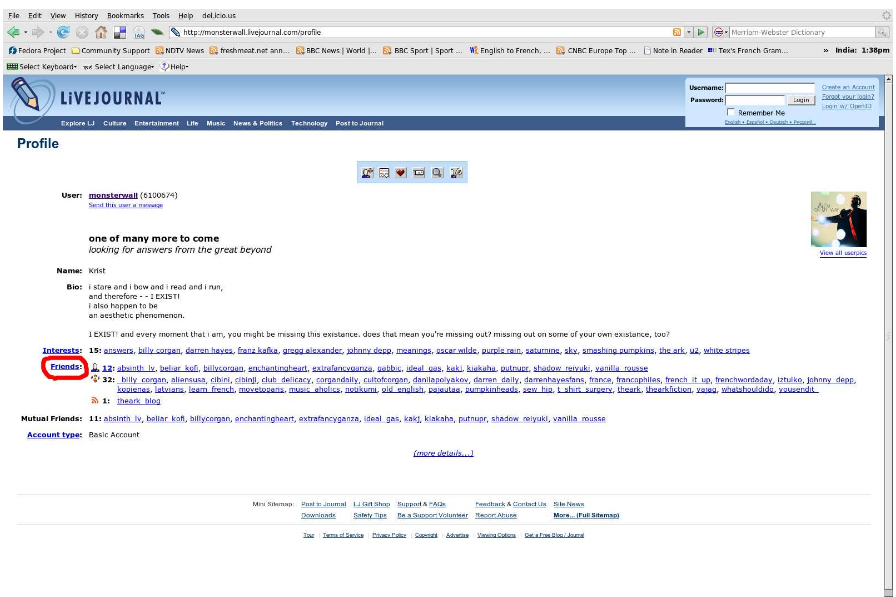
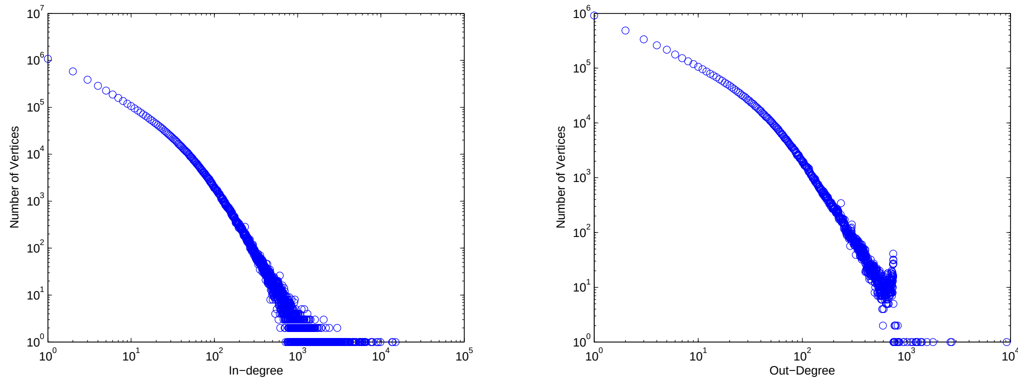


Figure 2: A typical user profile on LiveJournal (the *friends* section in the profile is marked in red).



(a) In-degree distribution. (b) Out-degree distribution.

Figure 3: Degree distributions in LiveJournal.

Let $G = (V, E)$ where $E \subseteq V \times V$ be a graph. If E consists of ordered pairs then the graph is known as a *directed graph* otherwise the graph is called *undirected*. The degree distribution, $P(k)$, of a graph is the probability distribution of the degree (k) of each vertex. In the case of directed graphs there are two degree distributions, *in-degree distribution* ($P_{in}(k)$) and the *out-degree distribution* ($P_{out}(k)$). Generally, social networks and the Internet have a peculiar degree distribution known as the *power-law*. Particularly, the degree distribution of such networks exhibits a power-law tail, $P(k) \sim k^{-\gamma}$. Figure (3) shows the in-degree and out-degree distributions for the LiveJournal graph. The parameter γ of the power-law for the LiveJournal graph is reported in Table (1) [15, 7, 16].

Along with the degree distribution generally one is also interested in the *mean degree* in a graph. If k_i is the degree of each vertex in an N vertex graph. Then the mean degree is simply defined as $\langle k \rangle = N^{-1} \sum_{i=1}^{N} k_i$ [16]. In social networks one is also interested to know whether the vertices with high degree tend to connect to other vertices with high degree. This characteristic of a network is captured by a parameter called the *assortative mixing coefficient*. If in a network, vertices with high degree tend to connect to other similar vertices then the network is said to exhibit *assortative mixing* [17, 18]. The assortative mixing coefficient is defined as follows: suppose the graph has M edges and at the end of the j^{th} edge it has two vertices having degree p_j and q_j then the assortative mixing coefficient is given as,

$$r = \frac{M^{-1} \sum_j p_j q_j - [M^{-1} \sum_j \frac{1}{2}(p_j + q_j)]^2}{M^{-1} \sum_j \frac{1}{2}(p_j^2 + q_i^2) - [M^{-1} \sum_j \frac{1}{2}(p_j + q_j)]^2}, \tag{1}$$

where $-1 \leq r \leq 1$. Generally social and collaborative networks show significant assortative mixing [17]. The assortative mixing coefficient for LiveJournal is reported in Table (1). A few other important properties are *average path length*, *radius*, and *diameter* of the graph. They are defined as follows: the *eccentricity* of a vertex v is the maximal shortest path distance between v and any other vertices. The radius of a graph is the minimum eccentricity across all vertices and the diameter is the maximum eccentricity across all vertices.

Unfortunately, these measures of a graph have a large computational complexity, thus generally they are computed on some randomly chosen subset of vertices. In Table (1) we give these parameters as computed by [15] on a random sample of 10,000 vertices of the LiveJournal graph. Another property of a graph is the size of the *strongly connected component* (or *SCC*) of a graph. An SCC of a graph is the set of vertices where every vertex in that set has a path (connected by edges) to every other vertex. However, we do not report the size of the SCC since the computation of SCC on a large graph is highly compute and memory intensive and with our limited resources it became more difficult. Later in Section (3) we generate smaller synthetic random graphs on which we compute the relative size of the SCC. If known, the size of the SCC is helpful for rumor spreading since if the selected vertex lies in this SCC then we are almost certain that the rumor would be heard by all the vertices in the SCC since all other vertices can be reached from this vertex. Social networks generally have a large SCC and thus forming a very suitable ground for spreading of rumors. Also social networks have smaller average path lengths and diameter as compared to the Web graph [6] and thus exhibit a small-world property.

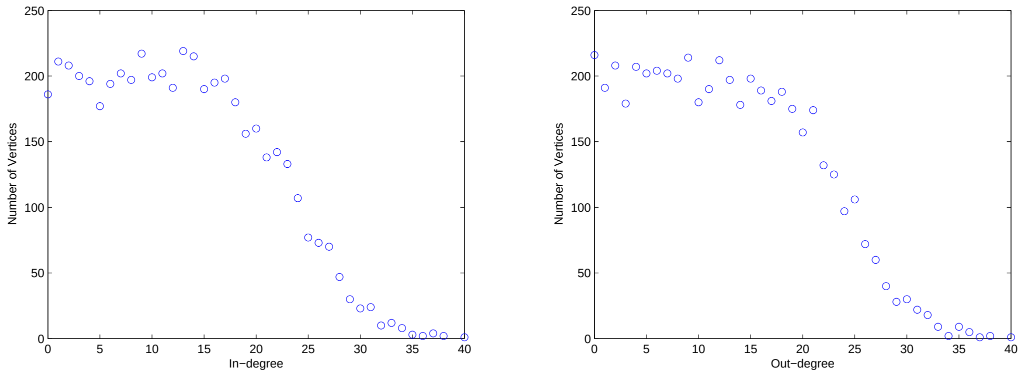
One property that is of interest and that reflects the local neighbourhood of a graph is the *clustering coefficient*. The clustering coefficient for a vertex v with R neighbours is the number of directed links that exist between the R neighbours and the number of possible links that can exist between these R neighbours ($R(R-1)$). The clustering coefficient (C) of graph is the average clustering coefficient of all the vertices. The clustering coefficient characterises how tightly all the vertices in a graph are linked to each other. In social networks the clustering coefficient is found to be orders of magnitude higher as compared to synthetically generated Erdős-Rényi or power-law graphs of the same size. This is natural since in social networks new friendships are formed based on common friends. Thus there is an high amount of *local* clustering in social networks. Generally, a higher clustering coefficient is typically observed in most social networks [16]. Again, the computation of the clustering coefficient is computationally intensive and in Table (1) we report the clustering coefficient computed in [15].

Parameter	Value
Number of vertices (N)	5,284,457
Number of edges (M)	77,402,652
Mean degree ($\langle k \rangle$)	29.2944
Power-law coefficients ($\gamma_{in}, \gamma_{out}$)	1.65, 1.59
Assortitative mixing coefficient (r)	0.5625
Clustering coefficient (C)	0.330
Average path length	5.88
Radius	12
Diameter	20

Table 1: Summary of structural properties of the LiveJournal social graph.

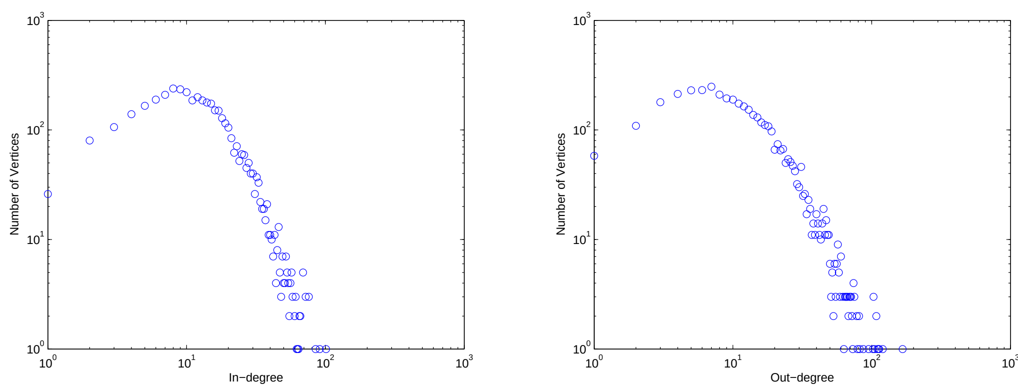
2.2 Rumor Model in Homogeneous networks

Now we describe the rumor spreading model. Let there be N vertices in a graph. Each of these vertices could be in three possible states, *stifler* or *spreader* or *ignorant*. This formulation of a rumor model follows the original terminology of [19, 10]. The three states of vertices refer to the different steps that could occur in rumor spreading dynamics. Ignorants are vertices that have not heard the rumor so far. Stiflers are vertices who have heard the rumor but have ceased spreading it. Likewise, spreaders are vertices that have heard the rumor and are actively spreading it. The densities (fraction) of these three types of vertices varies as the rumor spreads itself across a network. Let us denote these time varying densities by $i(t), s(t)$, and $r(t)$ for ignorants, spreaders, and stiflers respectively. These densities are normalized, thus,



(a) In-degree distribution. (b) Out-degree distribution.

Figure 4: Degree distributions of the synthetic Erdős-Rényi random graph.



(a) In-degree distribution. (b) Out-degree distribution.

Figure 5: Degree distributions of the synthetic power-law random graph.

$$i(t) + s(t) + r(t) = 1. \tag{2}$$

The dynamics of rumor spreading can be described as follows: a spreader converts an ignorant into a new spreader with probability λ. If a spreader encounters a stifler or a fellow spreader it becomes a stifler with probability α. This decay can be attributed to a mechanism of *forgetting* or because spreaders learn that the rumor has lost its *news value*. The spreader annihilation is directed, i.e. only the contacting spreader looses interest in spreading the rumor. The *mean-field rate equations* for this rumor spreading dynamics can be given as follows [10],

$$\frac{di(t)}{dt} = -\lambda\langle k\rangle i(t)s(t), \tag{3}$$

$$\frac{ds(t)}{dt} = \lambda\langle k\rangle i(t)s(t) - \alpha\langle k\rangle s(t)[s(t) + r(t)], \tag{4}$$

$$\frac{dr(t)}{dt} = \alpha\langle k\rangle s(t)[s(t) + r(t)], \tag{5}$$

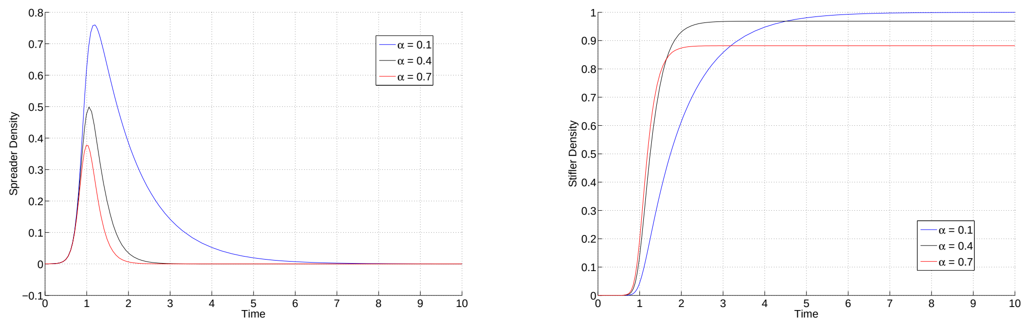
where $\langle k\rangle$ is the mean degree. The initial condition for rumor spreading are given following [10], as $i(0) = \frac{(N-1)}{N}$, $s(0) = \frac{1}{N}$, and $r(0) = 0$. These equations state that the number of spreaders at the current time instance (t) increase proportional to $\lambda\langle k\rangle$ and the fraction of spreaders and ignorants at the previous time instance $(t-1)$. While the annihilation process annihilates spreaders at a rate proportional to $\alpha\langle k\rangle$ and the number of spreaders and number of non-ignorants. We are particularly interested in the fraction of vertices that have learned the rumor or the *density of stiflers*, $r(t)$. However, to study the dynamics we also look at how the *spreader density*, $s(t)$, changes with time. Since this dictates the rate at which the rumor is spread and which intuitively depends on the properties of the underlying graph.

We numerically simulate the coupled differential equations (3), (4), and (5) using Matlab's differential equation solver [20]. In Figure (6) we show evolution of spreader and stifler densities when $N = 100$. The shapes of these graphs are typical, there is a peak in the spreader density since at some point of time there are large number of spreaders who have not encountered a fellow spreader or stifler. But then slowly the number of spreaders start to decrease and we get a bell-shaped curve of the spreader density for different values of parameter α. The stifler density on the other hand exponentially increases until after sufficient time every vertex in the graph has heard the rumor. In Section (3) we will give the actual simulation results of rumor spreading on a real graph (LiveJournal) and will show that spreader and stifler densities evolve in a very similar manner as depicted by the numerical simulation of the mean-field rate equations.

There are three different forgetting mechanisms or *annihilation rules* possible for the rumor spreading decay. Either the rumor spreading process dies proportional to the number of spreaders (*s-s interactions*) or to the number of stiflers (*s-r interactions*) or both (*s-s and s-r interactions*). Here we consider all the three different types of interactions and also compare them with each other. In the context of LiveJournal, the bloggers that are registered with LiveJournal can be thought of as the vertices in the graph. While the directed friendships (as described in Section (2.1)) that they make can be thought of as the edges in the graph. In this manner in the next section we study the rumor spreading problem in the context of real social networks.

3 Experimental Results

First, we initialize a vector of size N. We call this vector the *state vector*. This vector is used to track the states of all the vertices in a rumor spreading experiment. Throughout all the



(a) Spreader Density.

(b) Stifler Density.

Figure 6: Numerical simulation of the mean-field rate equations.

experiments we assume that $\lambda = 1$ without loss of generality. Each entry in the state vector indicates in which of the three states is the vertex. Now we start rumor spreading as follows: we choose two things a) value for $\alpha \in (0, 1)$ and b) a vertex uniformly at random from N (this is equivalent to choosing an entry uniformly at random from the state vector) and change its state to spreader. Then we start performing the rumor spreading iterations as described in Section (2.2). During each iteration we compute the spreader and stifler densities. We stop the iterations when the spreader and stifler densities have attained a steady state. For the same value of α we perform this procedure 5 times. Then we take an ensemble average of these 5 runs to obtain curves of spreader and stifler densities for this particular α value. Then we repeat the experiment with a different value of α. Lastly, we also change the annihilation mechanism and repeat the whole process.

In Figure (7) we show the evolution of spreader and stifler densities for all the three annihilation mechanisms with various α's for the LiveJournal social graph. Clearly, for smaller values of α we see that the spreader densities have a larger peak. Smaller values of α mean that the rumor is *forgotten* slowly. But with smaller values of α the final value of the stifler density can attain also decreases since the spreaders convert themselves to stiflers with a slower rate. On the other hand, for higher values of alpha we observe that the number of spreaders die out fast but the number of vertices that have heard the rumor at the end increases significantly. Also, between various annihilation mechanisms we observe that the peak of the spreader density and the final stifler densities for the *s-r* interaction are the highest. This is intuitive since once a vertex becomes a stifler it does not change its state and thus helping the spreading of rumor. On the other hand, unlike the *s-r* interaction, in *s-s* interaction only if two spreaders meet, an ignorant is created with a certain probability . Intuitively, this reduces the overall rate of rumor spreading since the states of the meeting vertices that result in the formation of an ignorant vertex are transient. In certain situations it can happen that the spreading of rumor terminates abruptly.

We now compare how rumor spreading performs within other types of synthetically generated directed random graphs and observe the differences. We generate one synthetic directed Erdős-Rényi random graph and one synthetic directed power-law random graph having properties described in Table (2). The power-law random graph was generated using the R-MAT algorithm [21]. There are two available implementations of this algorithm; PyWebGraph generator [22] and GTgraph [23]. The degree distributions of the Erdős-Rényi and power-law random graphs are shown in Figure (4) and Figure (5) respectively. Like described above we perform rumor

Parameter	Erdős-Rényi random graph	Power-law random graph
Number of vertices (N)	5000	5000
Number of edges (M)	62544	62545
Edge occurrence probability (p)	0.005	-
Mean degree ($\langle k \rangle$)	25.02	25.02
Assortitative mixing coefficient (r)	-0.81	0.83
Power-law coefficients ($\gamma_{in}, \gamma_{out}$)	-	2.59, 2.28
Relative size of the SCC	2×10^{-4}	0.812

Table 2: Summary of structural properties of the synthetic Erdős-Rényi and power-law random graph.

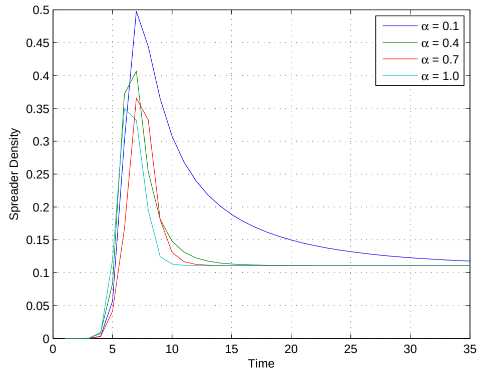
spreading runs on both the graphs and take an ensemble average over 100 runs. The results from these simulations are shown in Figure (8) and Figure (9). Clearly, it can be observed that the peak spreader density and the final stifler densities attained in the Erdős-Rényi random graph is much smaller than the power-law random graph. This can be attributed to the preferential attachment mechanism that generates a power-law random graph. Even though the densities in the power-law random graph are higher they are not as high as the densities observed in the LiveJournal graph. Since the LiveJournal graph has very high clustering coefficient and exhibits a small world phenomenon. Admittedly, LiveJournal is more susceptible to spreading of rumors than other graphs. Furthermore, as all social networks show similar characteristics as LiveJournal, it could be suggested that social networks are more susceptible to rumor spreading than power-law or Erdős-Rényi random graphs.

3.1 Implementation details

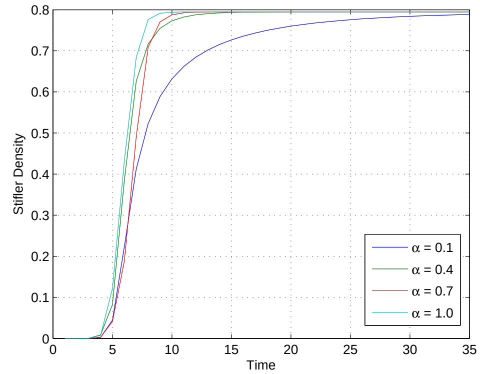
All the experiments were performed on a IBM Lenovo T60 Laptop having a Intel Core2 Duo processor at 2 Ghz and with 1GB of memory. Python 2.5 [24] was used for coding and MySQL 5.0 [25] was used to store the graphs as a list of edges. We indexed the table holding the edge list on the *start* and *end* vertices of an edge. This enables use to traverse the graph forward an backward with using only simple `SELECT` queries. The total disk size of the database including the index size was about 2.1 GB. The rumor spreading experiment for LiveJournal graph took around three days to complete. But for the Erdős-Rényi and power-law graphs the simulation took about one hour. The massive size of the LiveJournal graph and the limited resources available justify the time taken for this simulation.

4 Conclusions and Future Work

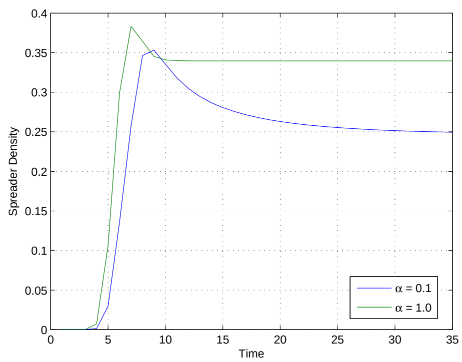
General behavior of the rumor spreading process was outlined in this report. Our simulation shows that the behavior of rumor spreading process is considerably different for social networks as compared to Erdős-Rényi and power-law graphs. The peak spreader and final stifler densities are significantly higher for the LiveJournal social graph. This clearly suggests that social networks are more susceptible to rumors than the other networks considered in this report. Furthermore, evolution of spreader and stifler densities on LiveJournal closely resemble the numerical simulation of the mean-field rate equations described in Section (2.2). In the annihilation mechanisms that we considered a spreader turns itself into a stifler only after meeting another spreader or sti-



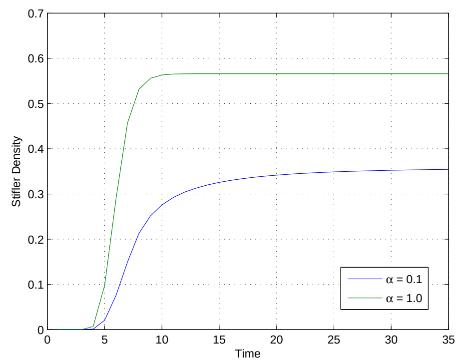
(a) Spreader density for *s-s and s-r* interaction.



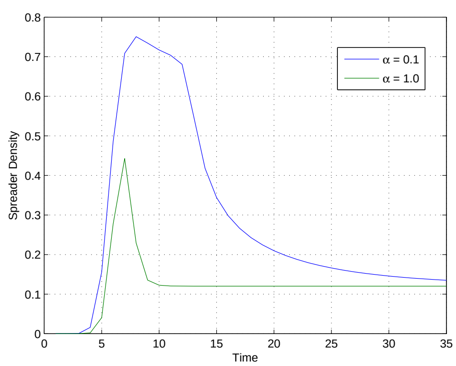
(b) Stifler density for *s-s and s-r* interaction.



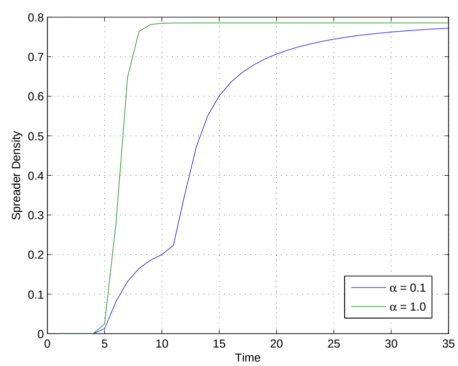
(c) Spreader density for s-s interaction.



(d) Stifler density for s-s interaction.

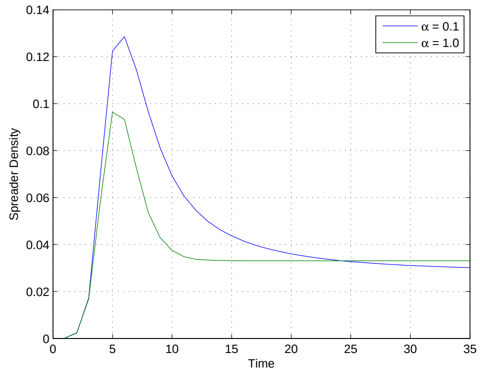


(e) Spreader density for s-r interaction.

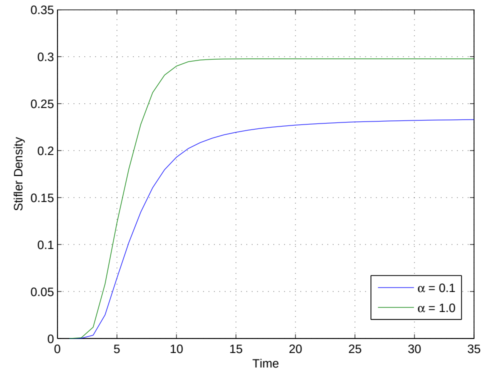


(f) Stifler density for s-r interaction.

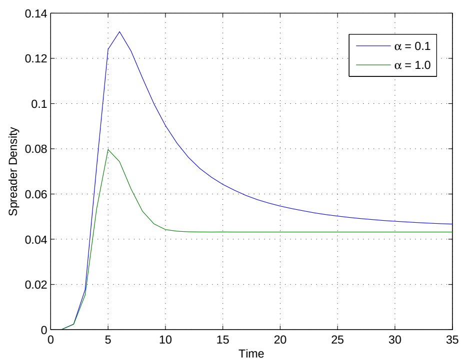
Figure 7: Evolution of spreader and stifler densities against various annihilation mechanisms with various α's in the LiveJournal social graph.



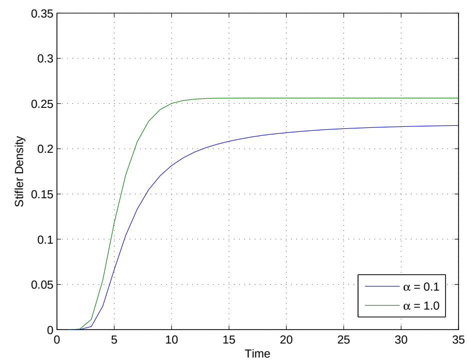
(a) Spreader density for *s*-*s and s*-*r* interaction.



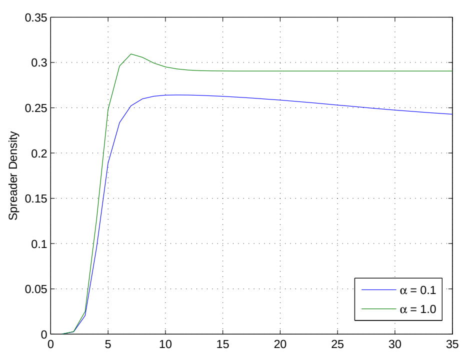
(b) Stifler density for *s*-*s and s*-*r* interaction.



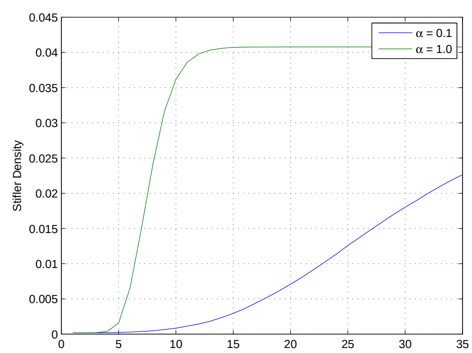
(c) Spreader density for *s*-*s* interaction.



(d) Stifler density for *s*-*s* interaction.

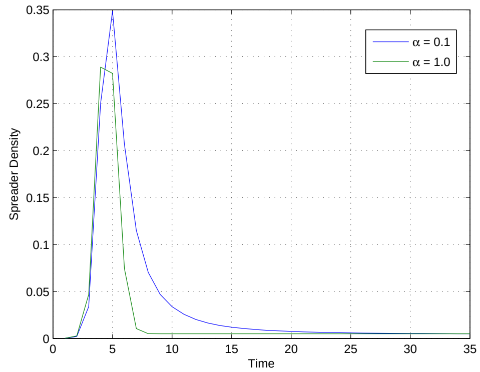


(e) Spreader density for *s*-*r* interaction.

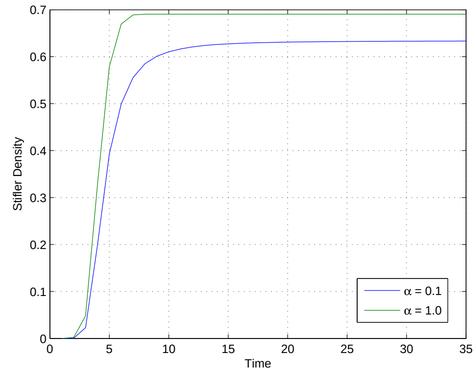


(f) Stifler density for *s*-*r* interaction.

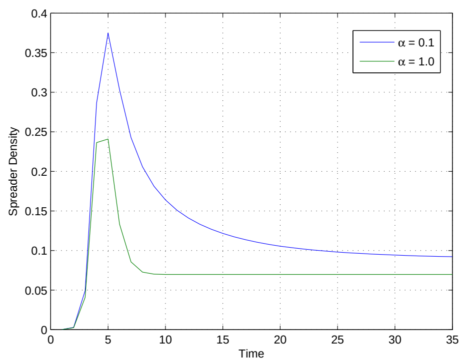
Figure 8: Evolution of spreader and stifler densities against various annihilation mechanisms with various α's in the Erdős-Rényi random graph.



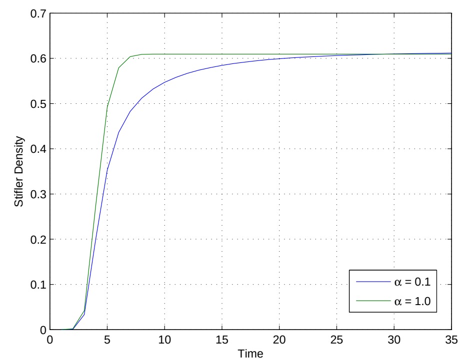
(a) Spreader density for *s*-*s and s*-*r* interaction.



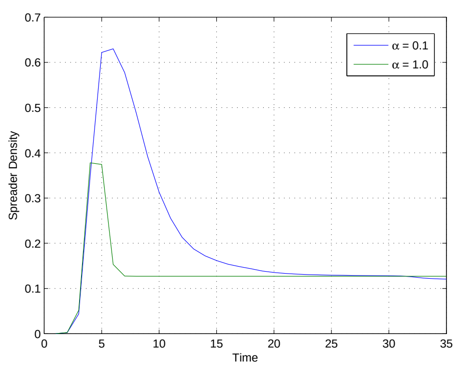
(b) Stifler density for *s*-*s and s*-*r* interaction.



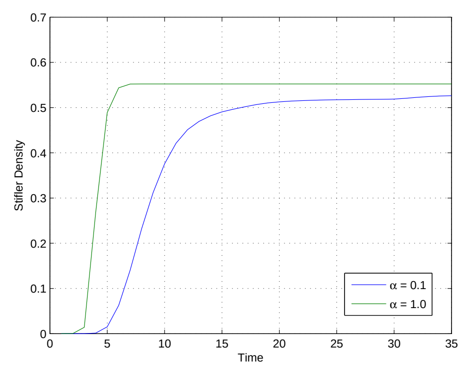
(c) Spreader density for *s*-*s* interaction.



(d) Stifler density for *s*-*s* interaction.



(e) Spreader density for *s*-*r* interaction.



(f) Stifler density for *s*-*r* interaction.

Figure 9: Evolution of spreader and stifler densities against various annihilation mechanisms with various α's in the power-law random graph.

fler. In terms of future work, there is another annihilation mechanism which was not considered in this report. This method suggests that a good representation of the rumor spreading model would be when a vertex, with a certain probability, converts itself to a stifler even though it has not encountered anyone. This represents a more natural process of *forgetting* to forward a rumor to ones neighbours. The details of this process are outlined in [14]. Furthermore, due to the high computational complexity involved in estimating parameters like the clustering coefficient there is also a need to further investigate the introduction of new parameters which have low computational complexity but are able to capture similar characteristics of the network. This becomes relevant since lately large social networks have been developed on the Web.

Acknowledgement

We would like to thank Alan Mislove and Sheila Kinsella for giving us the social network datasets. We would also like to thank Prof. Martin Hasler for advising this project.

References

[1] P. Erdős and A. Rényi, "On random graphs," *Publ. Math. Debrecen*, vol. 6, no. 290, 1959. 1, 2

[2] "Orkut," http://www.orkut.com. 1, 2

[3] "Facebook," http://www.facebook.com. 1, 2

[4] "hi5," http://hi5.com. 1

[5] "Livejournal," http://www.livejournal.com. 1, 2

[6] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the Web," *Computer Networks*, vol. 33, no. 1-6, pp. 309–320, 2000. 1, 5

[7] R. Albert and A. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002. 1, 4

[8] "DarkReading," http://www.darkreading.com/document.asp?doc_id=141761, Google's Orkut Social Network Hacked. 1

[9] "PayPerPost," http://payperpost.com/. 1

[10] Y. Moreno, M. Nekovee, and A. Pacheco, "Dynamics of rumor spreading in complex networks," *Physical Review E*, vol. 69, no. 6, p. 66130, 2004. 2, 5, 7

[11] Y. Moreno, J. Gómez, and A. Pacheco, "Epidemic incidence in correlated complex networks," *Physical Review E*, vol. 68, no. 3, 2003. 2

[12] D. Zanette, "Critical behavior of propagation on small-world networks," *Physical Review E*, vol. 64, no. 5, 2001. 2

[13] ——, "Dynamics of rumor propagation on small-world networks," *Physical Review E*, vol. 65, no. 4, 2002. 2

[14] M. Nekovee, Y. Moreno, G. Bianconi, and M. Marsili, "Theory of rumour spreading in complex social networks," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 1, pp. 457–470, 2007. 2, 13

[15] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 29–42, 2007. 2, 4, 5

[16] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003. 4, 5

[17] ——, "Assortaive Mixing in Networks," *Physical Review Letters*, vol. 89, no. 20, 2002. 4

[18] ——, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, 2003. 4

[19] D. Daley and D. Kendall, "Epidemics and Rumours," *Nature*, vol. 204, no. 4963, p. 1118, 1964. 5

[20] "MATLAB," http://www.mathworks.com. 7

[21] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-MAT: A Recursive Model for Graph Mining," *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004. 8

[22] "PyWebGraph Generator," http://pywebgraph.sourceforge.net. 8

[23] "GTgraph," http://www.cc.gatech.edu/~kamesh/GTgraph/, A Suite of Synthetic Graph Generators. 8

[24] "Python," http://www.python.org. 9

[25] "Mysql," http://www.mysql.com. 9