MIS Quarterly

RESEARCH NOTE

EMERGENCE OF POWER LAWS IN ONLINE COMMUNITIES: THE ROLE OF SOCIAL MECHANISMS AND PREFERENTIAL ATTACHMENT

Steven L. Johnson
Fox School of Business, Temple University, Philadelphia, PA 19122-6083 U.S.A. {steven@temple.edu}

Samer Faraj
Desautels Faculty of Management, McGill University, Montreal, Quebec H3A 1G5 CANADA {samer.faraj@mcgill.ca}

Srinivas Kudaravalli
HEC Paris,1, rue de la Liberation, 78351 Jouy en Josas Cedex FRANCE {kudaravalli@hec.fr}

Appendix A

Detailed Description of Research Method

As summarized in Table A1, this appendix provides a detailed description of the research method used to test the proposed hypotheses. The first major step was to collect observational and trace data from online communities. The observational data supported creation of a descriptive model of online community formation. The trace data encompasses the first 12 to 15 months of communication in 28 online communities that were randomly selected into a model specification sample and a model testing sample. In the next major step, the descriptive model was turned into a formal model of online community formation and then implemented in R. The first 12 to 15 months of data from the online communities in the model specification sample provide key simulation parameters (e.g., participant arrival and departure rates).

Table A1. Research Method

Activity	Steps	Description
Collect Data	Observe Online Communities	Through observation of online communities identify behaviors, rules, and tendencies guiding communication network formation.
	Prepare Descriptive Model	Complement observation data with existing theory and empirical research to prepare descriptive model of online community network formation.
	Collect Trace Data	Collect 15 months of message history for multiple online communities and randomly split into model specification sample and model testing sample.
Develop Simulation	Create Formal Model	Document formal rules and assumptions. Identify functional forms for agents and mechanisms.
	Calculate Input Parameters	Use model specification sample to calculate input variables for simulation.
	Code Simulation	Implement formal model as an agent-based simulation in R.
Run Simulation	Generate Data	Generate simulation data by varying parameters associated with each hypothesis. Run single mechanism setting for test of H1 (n = 40 per individual mechanism) and run 50 Monte Carlo mechanism combinations (n = 20 per mechanism combination) for test of H2 (total n = 20 * 50 = 1,000).
	Validate Simulation Output	Compare model specification sample and simulation data to validate simulation model generates data in empirical range of interest.
	Calibrate Simulation Parameters	Prepare analysis sample of networks with similar structural characteristics as model specification sample.
Model Testing	Calculate Outcome Measures	Calculate outcomes in the model testing sample and the calibrated simulation output.
	Analyze Model	Compare outcomes to test hypothesis.

In the third major step of the research method, simulation data was generated to test each of the proposed hypotheses. First, data was generated with parameters specific to each hypothesis. Second, the entire set of generated data was compared to the model specification sample to establish that the simulation encompasses an appropriate range of networks. Next, the simulation model was calibrated to create an analysis sample. The final major step was to analyze the similarity of the model testing sample and the analysis sample. The estimated power law distribution was calculated for all of the networks in both samples. Then, two-tailed ANOVA tests were performed to test each hypothesis. All four major steps are described in more detail in this appendix.

Collect Data

This paper studies communication networks formed via participants in online communities supported by threaded discussion forums. The first major activity in the research method was to collect observation data and trace data to prepare a descriptive model of the network formation process. Several hundred threaded discussion forums were informally observed. These online communities were identified using Yahoo! Search for the strings "Powered by vBulletin" and "software" to find technology-related discussion forums that share a common technology infrastructure. The sample was further reduced to include only online communities where a full archive of the initial 15 months of posting history was available. An automated program was used to gather the first 15 months of communication for 28 online communities randomly selected.

Table A2. Model Specification and Model Testing Subsamples

Website	Subssample	Inception	Participants	Messages	Threads
forums.3dcart.com	Specification	2/24/06	78	950	254
bid-alot.com/forum	Test	8/11/04	62	1492	291
bibleworks.com/forums	Test	4/1/04	345	3388	647
forums.builtbp.com	Test	2/8/06	140	247	114
archive.burningsea.com/forums	Test	10/25/01	214	2156	382
codenewbie.com/forum	Specification	5/6/02	248	5596	893
forum.conceiva.com	Specification	12/24/07	84	434	133
cruisersforum.com/forums	Test	2/24/03	297	3319	686
developers.evrsoft.com/forum	Test	9/5/02	959	8996	1425
forums.foxitsoftware.com	Test	6/27/08	699	3429	895
fsdeveloper.com	Test	6/1/04	281	5758	953
gamefileforums.com/forums	Test	1/31/05	1112	5703	1047
forums.hostrocket.com	Specification	5/10/02	389	8022	1388
jpsoft.com/forums	Test	5/20/08	214	4074	851
forum.k-billing.com	Specification	9/11/06	127	1181	285
libertystreet.com/forums	Specification	11/5/05	131	2279	332
forums.mxhub.com	Test	5/12/01	85	519	172
npowersoftware.com/forums	Test	9/4/03	65	434	106
programmersresource.com/forum	Test	5/21/03	615	4409	1390
siginetsoftware.com/forum	Specification	5/16/06	222	1238	137
stormlabstuff.com/board	Test	6/1/07	106	1560	286
forums.swordsearcher.com	Specification	5/16/06	65	927	168
teamd86.com	Test	11/27/02	86	3880	552
bb.turtlesoft.com	Test	9/13/06	26	280	83
vintage-computer.com/vcforum	Test	4/27/03	266	5184	883
vjforums.com	Test	4/8/02	830	17057	2110
weathergraphics.com/forum	Specification	11/23/03	141	1763	493
forums.winxpcentral.com	Test	7/29/01	453	8929	2233

Online Community Sample

Online communities in Table A2 were randomly assigned to a model specification and model test subsample. Each online community was assigned a random number using the Microsoft Excel RAND() function. The list was then sorted in ascending order with the one third (the first 9 groups) assigned to the model specification subsample and the remaining 19 groups assigned to the model test subsample. The original online community URL is listed. Since the research data was collected, some online communities have moved to a new URL, migrated to a different software platform, or closed altogether.

Descriptive Model of Online Community Threaded Discussion Board

This paper simulates the formation of online community communication networks formed via threaded discussion boards. The rules and assumptions described in Table A3 were used to model the formation of and communication within a threaded discussion board. The primary entities in modeling an online community threaded discussion board are participants, messages, and threads.

Table A3. Descriptive Model

Participants

Participants are limited to these actions:

- Enter system
- Post a message that starts a new thread
- Post a message that adds to an existing thread
- Exit system

All participants share these behaviors:

- New participants may arrive at any time
- Any active participant can post a message to start a new thread
- Any active participant can post a message in response to the an open thread
- Active participants may depart the system at any time

To facilitate modeling, participants are assumed to behave in this way:

- Immediately upon arrival a participant posts a message that starts a new thread
- After a participant exits the system they may not reenter
- No participants are present in the system when the simulation begins
- Participants will have between 1 and a preset maximum number of opportunities to post per turn

Messages

Rules for messages:

- Every message belongs to one and only one message thread
- All message are treated as responses to the message starting the message thread

Threads

Rules for threads:

- New message threads may be created with a new message by already active or newly entering participants at any time
- Any active or newly entering participant may respond to any open message thread

As an approximation of observed preferences for newer threads, this assumption is made:

- A message thread is made inactive (closed) 8 clicks (days) after it is created[†]

[†]Across all of the model specification subsample, 82% of the message threads received all of their responses within 8 days; 85% received all but one message, and 91% received all but two messages.

Develop Simulation

The next major step of the research method was to develop the simulation. The descriptive model developed in the first step was further developed as a formal network formation model. Where appropriate, the first 12 to 15 months of data from model specification sample provided

key simulation parameters. For example, the first year of data was used to estimate participant arrival rates and the first 15 months of data to estimate participant departure rates. By expressing key behaviors and rules in mathematical formulas, this detailed formal model provided the basis for implementing the agent-based model of network creation in the programming language of R.

Formal Model for Online Community Formation

Because no existing theoretical basis was found to develop arrival and departure functions, the new participant entry and participant departure functions were based on empirically derived formulas calculated from the model specification subsample. New participant entry was modeled as a function of time and active participant departure as a function of individual tenure.

To analyze new participant entry, a cumulative distribution function was calculated based on when participants posted their first message. It was found that the cumulative distribution functions for the online communities in the model specification subsample closely fit a linear model (average $R^2 = 0.88$; s.d. $= 0.09$; $n = 9$).

To analyze participant departure, the cumulative distribution function was calculated based on the length of time between a participant's first message and their final message in the first year. Any participant who posted a message during months 12–15 was considered as remaining active. The best fitting model for the cumulative distribution function for participant departure was found to be a logarithmic function (average $R^2 = 0.96$; s.d. $= 0.03$; $n = 9$).

Figure A1 shows an example cumulative distribution function for arrival and departure. The observed arrival pattern is straightforward: new participants arrive in a relatively steady pace with minor variations. The departure function shows that although a high percentage of participants depart between one day and one month of arrival (in Figure A1 almost 70% and 85%, respectively), if a participant remains active beyond approximately one month, their marginal probability of departure decreases. That is, participants who have been participating for a longer period of time are less likely to discontinue participating than those who have been participating for a shorter period of time.

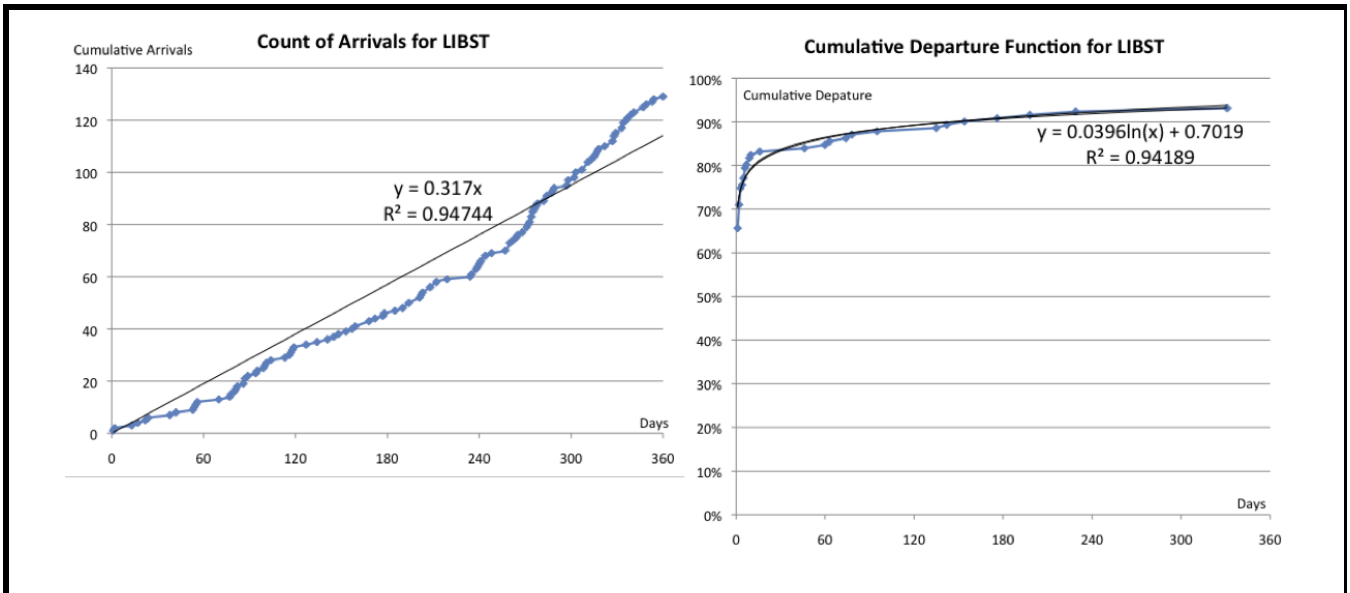


Figure A1. Example Participant Arrival and Departure Functions

Selecting Network Generation Mechanisms

The primary research question for the paper is to identify network generation mechanisms that lead to the power law distributions observed in online communities. A critical step in the research process is identifying the most relevant mechanisms to test. We selected mechanisms that meet one of the following two criteria: (1) they are relevant mechanisms known to create power law distributions in other settings or (2) they are mechanisms identified in previous studies of online communities. The latter is described in the body of the paper, the former below.

Table A4. Generation Mechanisms for a Power Law Distribution

Dominant Domain	Mechanism	Relevance to Online Community Formation
Biology	Contagion bursts	Is based on contagions and epidemics; no parallel exists in online community formation.
Biology	Hierarchical modularity	A model of cellular level growth through cell fission; online community growth based on arriving and departing participants.
Biology	Interacting fractals	Is based on fractal structure of food webs and predators; no parallel exists in online community formation.
Economics	Niche proliferation	Based on effects of scale in markets; no parallel exists in online community formation.
Economics	Random walk	Describes distribution of losses before a finite resource depleted in a stochastic process; no parallel exists in online community formation.
General	Combination theory	Combination theory is not, by itself, a generation mechanism; instead, it suggests that power laws can form through the combination of complex subtasks. A blended model of network formation may be consistent with combination theory but does not directly test it.
General	Event bursts	Describes prioritization of activity, such as bursts of communication; no empirical evidence for existence in online community formation.
General	Interactive breakage theory	Interactive breakage theory is not, by itself, a generation mechanism; instead, it suggests that power laws can form through positive feedback loops. A blended model of network formation may be consistent with interactive breakage theory but does not directly test it.
General	Spontaneous order creation	Spontaneous order creation is not, by itself, a generation mechanism; instead, it suggests that power laws can form through positive feedback loops. A blended model of network formation may be consistent with spontaneous order creation but does not directly test it.
Physical	Irregularity generated gradients	Is based on autocatalytic processes; no parallel exists in online community formation.
Physical	Phase transition	Is based on autocatalytic processes; no parallel exists in online community formation.
Physical	Self-organized criticality	A physical theory related to the effects of gravity; no parallel exists in online community formation.
Physical	Surface-volume law	Based on relationship between 2D surface and 3D volume; no parallel exists in online community formation.
Social	Least effort	Applies in situations like online communities where some nodes have varying connection costs.
Social	Preferential attachment	Applies in situations like online communities where new entrants choose among existing entrants for connections

To identify relevant mechanisms, we started with the list identified in Andriani and McKelvey's (2009) comprehensive review (see Table A4). As a next step, building on theoretical and empirical work on online communities, we developed criteria as to what characteristics those mechanisms would have. Based on literature on online communities (Butler 2001; Faraj and Johnson 2011; Preece 2000; Ren et al. 2012) , we selected only the social mechanisms. That is, the mechanisms that are likely to build member attachment and participation in online communities.

Our fundamental argument is that models borrowing from areas such as heat transfer, gravity, phase transition, contagion, fractals, or random walk may be useful descriptive metaphors and empirical descriptions of a static network, but they cannot be considered relevant as explanatory mechanisms for the highly social process involving human actors interacting in online communities—as is now well established in the above cited literature. Thus, this paper is not focused on proving or disproving the families of models coming from other scientific domains. Instead, the focus is on evaluating socially relevant mechanisms.

Power Law Distribution Generation Mechanisms

In the implementation of preferential attachment, the likelihood of selecting an agent to respond to is in proportion to the number of previous responses they have received (their in-degree). The linearized chord diagram (LCD) implementation (Kolaczyk 2009, pp. 173-175) of the Barabási and Albert (1999) model of preferential attachment was used. The LCD implementation disambiguates the original Barabási and Albert model in regard to both starting conditions for a new network, as well as selection for new connections.

In the implementation of least efforts, every active agent was assigned a relative weight representing the ease with which they may be responded to. This parallels the logic of least efforts that, just as there are shorter words in a vocabulary that are easier to use than longer ones, within an online community, there are some people who consistently create threads that are easier to respond to than others. The probability was assigned once for each agent and remains constant for the entire simulation of that communication network formation. In determining who a focal agent will respond to, an agent was selected randomly, weighted by the least effort weight. If the selected agent has more than one active thread, the active thread that was created first was responded to.

In the implementation of direct reciprocity, the likelihood of selecting an agent to respond to was based on prior dyadic communication history. First a weight was calculated for each potential agent as follows, where r_{ij} was the number of previous messages from the focal agent (i) to the potential agent (j) and r_{ji} was the number of messages from the potential agent to the focal agent:

- If r_{ij} is 0 and $r_{ji} > 0$ then set weight to 2
- If r_{ij} is 0 and r_{ji} is 0 then set weight to 1
- If $r_{ij} > 0$ and $r_{ji} > 0$ then set weight to $(1 + (r_{ij} - r_{ji}) / (r_{ij} + r_{ji}))$

The intuition of this formula is that when two participants have not interacted, they start with a weight of 1. When a focal agent has not yet reciprocated a tie from an alter, motivation is greatly increased (weight = 2); when a focal agent has unreciprocated ties, motivation decreases in proportion to the number of unreciprocated ties. Once the weights were calculated for an agent and all potential alters, an alter was randomly selected weighted by the calculated values. If the selected agent has more than one active thread, the active thread that was created first was responded to.

With indirect reciprocity, an agent's likelihood of creating new links is determined by its previous history of posts and replies to those posts (Faraj and Johnson 2011). The formula was calculated based on the ratio of an agent's in-degree divided by the agent's out-degree. If an agent has received many replies, but not made many posts themselves, their in-degree will exceed their out-degree. The ratio of the in-degree to the out-degree will be greater than one and an agent will have an above-average probability of creating new ties. If an agent has made many replies to others, but not yet received as many replies to their own threads, the situation was reversed. The ratio will be less than one and that agent's probability of creating new ties will be less than average. In either case, when agents create new ties, they select them at random from all other active agents in the network. In the starting condition, when an agent has not yet posted any replies (e.g., their out-degree is zero), they were assigned the average propensity to form new ties.

Calculate Input Parameters

The shaded values in Table A5 were used as simulation input parameters.

Table A5. Key Statistics From Model Specification Subsample

Website	Thread Depth	Departure Parameter (b)	Departure Parameter (Int)	Participants	Messages	Threads
forums.3dcart.com	3.740	0.07	0.18	78	950	254
codenewbie.com/forum	6.279	0.06	0.49	248	5569	887
forum.conceiva.com	3.273	0.07	0.51	84	432	132
forums.hostrocket.com	5.782	0.09	0.23	389	7996	1383
forum.k-billing.com	4.144	0.08	0.38	127	1181	285
libertystreet.com/forums	6.865	0.04	0.70	131	2279	332
siginetsoftware.com/forum	9.074	0.05	0.60	222	1225	135
forums.swordsearcher.com	5.518	0.05	0.57	65	927	168
weathergraphics.com/forum	3.551	0.09	0.22	141	1747	492
Average (n = 9)	5.358	0.067	0.431	165.0	2478.4	452.0

Code Simulation

The formal model was implemented in computer code; a pseudocode representation is provided below. The computational model was implemented in the statistical package R (R Development Core Team 2009). Additional packages used include igraph (http://igraph.sourceforge.net), MASS (http://www.stats.ox.ac.uk/pub/MASS4/), and sampling (Matei and Tillé 2006). Consistent with best practices of simulation development, common random number queues were used across simulation runs as a variance reduction technique (Law and Kelton 2000). The first author programmed the simulation and the second author reviewed the code. The portion of the pseudocode shown in Figure A3 demonstrates overall flow control.

```
# comment:  model of online community communication network formation

Set arrival schedule for each agent
Set departure schedule for each agent
Set LeastEffortsWeights for each agent to a random value

FOR each click in simulation period

 Set agents arriving this click to active status
 Create a new thread this click for each newly arrived agent
 Set threads expiring this click to expired

 FOR each agent with status of active

  Set network formation mechanism for this agent for this click

  IF network formation mechanism is Indirect Reciprocity THEN
   Set postperturn based on agent indegree / outdegree
  ENDIF

  FOR each postperturn up to maxpostperturn
   Call network formation function with active agent as "from"
  END FOR

 END FOR

 Set agents departing this click to inactive status

END FOR

Save message history to file
```

Figure A3. Overall Flow Control

The portion of the pseudocode shown in Figure A4 describes the network formation function.

```
# comment:  network formation function

IF network formation mechanism is Preferential Attachment THEN
 # comment:  randomly select reply 'to' weighted by in-degree
 Set RandomThread to thread containing a randomly selected post
 Set "to" agent as agent that started RandomThread
 IF any threads started by "to" agent are active THEN
  Post to earliest active thread of 'to' agent
 ENDIF
ENDIF

IF network formation mechanism is Least Efforts THEN
 # comment:  randomly select reply 'to' weighted by ease of replying to
 Set agentlist to all agents with active threads
 Set 'to' agent as random selection from agentlist using LeastEffortsWeights
 Post to earliest active thread of 'to' agent
ENDIF

IF network formation mechanism is Direct reciprocity THEN
 # comment:  randomly select reply 'to' weighted by past history
 Set agentlist to all agents with active threads
 FOR all agents in the agentlist
  Set Rij to number of posts from focal agent to agent in agentlist
  Set Rji to number of posts to focal agent by agent in agentlist
  IF Rij is zero and Rji > 0 THEN Set weight to 2
  ELSE IF Rij + RJI is zero THEN Set weight to 1
  ELSE Set weight to 1 + (Rij – Rij) / (Rij + Rji)
 ENDFOR
 Set 'to' agent as random selection from agentlist using weights
 Post to earliest active thread of 'to' agent
ENDIF

IF network formation mechanism is Indirect Reciprocity THEN
 # comment:  randomly select reply 'to' agent
 Set agentlist to all agents with active threads
 Set 'to' agent as random selection from agentlist
 Post to earliest active thread of 'to' agent
ENDIF
```

Figure A4. Network Formation Function

In a blended model, multiple mechanisms are present during network formation. Weights were identified for each network formation mechanism that remain constant for that simulation run. The weights to test were chosen with a Monte Carlo method (see Law and Kelton 2000). A random number was chosen for each of the four mechanisms and then a percentage of total (0%–100%) calculated for each mechanism. For each simulated day, an active agent is assigned a single network formation mechanism based on the weights for that simulation run.

Prior to running the simulation code, the list of random number seeds was selected by randomly selecting numbers between 1 and 10,000 using the R command:

```
sort (trunc (runif (10, 1, 10000)))
```

This command returns 10 randomly selected values from a uniform distribution, converts them to an integer value via truncation and then sorts the list in ascending order. The pseudocode shown in Figure A5 describes how the random number seeds are incorporated into the simulation model.

```
# comment:  blended model algorithm (Monte Carlo)
Set RandomNumberList to list of random number seeds

FOR all seeds in RandomNumberList
 Set Random1 to randomly selected number between 0 and 1
 Set Random2 to randomly selected number between 0 and 1
 Set Random3 to randomly selected number between 0 and 1
 Set Random4 to randomly selected number between 0 and 1
 Set TotalRandom = Random1 + Random2 + Random3 + Random4
 Set PreferentialAttachmentWeight = Random1 / TotalRandom
 Set LeastEffortsWeight = Random2 / TotalRandom
 Set DirectReciprocityWeight = Random3 / TotalRandom
 Set IndirectReciprocityWeight = Random4 / TotalRandom

 [ ...  call flow control code ...  ]

 # Where the command reads:
 #  Set network formation mechanism for this agent for this click
 # Randomly select mechanism weighted by the mechanism weights
 #  PreferentialAttachmentWeight, LeastEffortsWeight,
 #  DirectReciprocityWeight and IndirectReciprocityWeight

ENDFOR
```

Figure A5. Incorporating Random Number Seeds into the Simulation Model

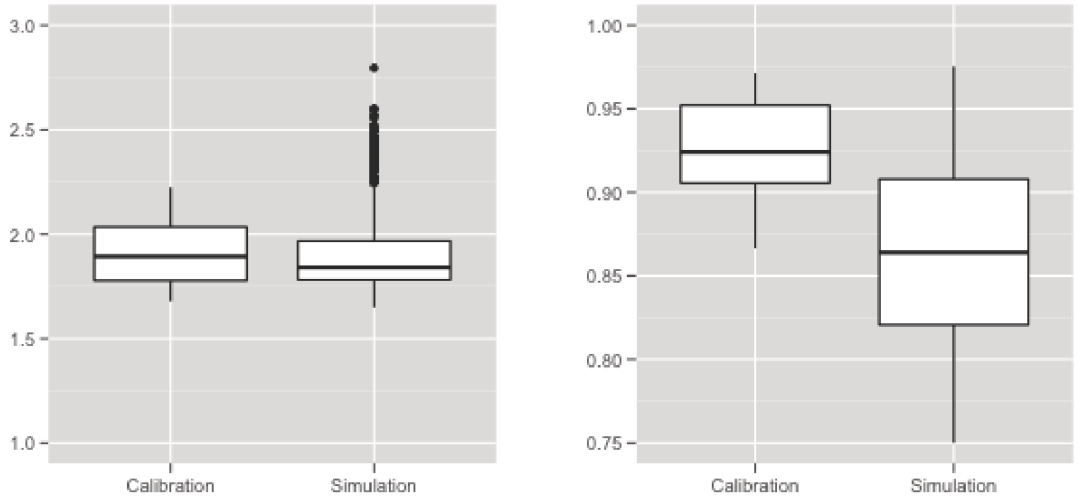
Run Simulation

Generate Simulation Data

Each set of simulation parameters was run with multiple random number queues. A simulation run was comprised 365 iterations, simulating the first 365 days of an online community formation. An active agent could create zero to three responses to existing messages and could initiate zero or one new message threads per iteration. The ability to create multiple messages per period is consistent with actual participant behavior and also assisted in creating networks of similar size to the model specification sample.

Validate Simulation

A comparison of simulated data to observed data validates that the simulation generates results consistent with reality. As demonstrated visually in Figure A6, the simulated data covers the range of observed data. It is unsurprising that some simulated networks were out of range and encouraging that others were a closer fit. In this way we can establish what mechanisms are most consistent with the emergence of power laws in online communities and can also provide confidence that our computational model provides a faithful simulation.



Estimated Power Law Exponent (*b*) and fit (R^2) for Calibration Sample (n=9) and Simulation Runs (n=1040)

Figure A6. Model Validation

Table A6. Additional Calibration Variables From Model Specification Sample

Website	Graph Density	Clustering Coefficient
forums.3dcart.com	0.042	0.274
codenewbie.com/forum	0.016	0.204
forum.conceiva.com	0.014	0.021
forums.hostrocket.com	0.012	0.155
forum.k-billing.com	0.012	0.027
libertystreet.com/forums	0.009	0.038
siginetsoftware.com/forum	0.006	0.009
forums.swordsearcher.com	0.046	0.269
weathergraphics.com/forum	0.021	0.160
Minimum	0.006	0.009
Maximum	0.046	0.274

Calibrate Simulation

The calibration of a simulation model provides confidence in finding robust configuration to compare against the model testing sample (Sanchez and Lucas 2002). Two methods were employed to calibrate the simulation. First, to simulate networks of matching size and complexity, the target number of message per threads from subsample A was used (5.4) as a target value. Simulation runs resulting in an average number of messages per thread +/- 10% of this target were retained. Simulation runs falling below the target thread percentage range were dropped because it is an indication that network formation mechanisms were not being consistently triggered. In essence, the mechanisms did not have enough power compared to the modeling assumption that a new entrant starts a new thread upon arrival. Simulation runs where the target thread percentage was overshot were dropped because it is an indication that the rule to keep thread length in balance is overwhelmed by the mechanisms. Table A6 shows the additional calibration variables (thread depth appears in Table A4). Table A7 summarizes the relationship between the model specification sample, model testing sample, and analysis sample.

Table A7. Summary of Samples		
Major Activity	**Sample**	**Sample Size**
Prepare Descriptive Model	Observation Sample	Full message history for first 12–15 months of 28 online communities.
	Model Specification Sample	n = 9; randomly selected third of observation sample
	Model Testing Sample	n = 19; remainder of observation sample
Run Simulation	Generate Data	For H1 test: 40 runs with each single mechanism setting (total n = 40 * 4 = 160) For H2 test: 50 Monte Carlo mechanism combinations with 20 runs per mechanism combination: total n = 20 * 50 = 1,000.
	Validate Simulation Output	Comparison of model specification sample (n = 9) and simulation runs (n = 1,160)
Model Testing	Analyze Model	For H1 test: analysis sample (n = 40) compared to model testing sample (n = 19) For H2 test: analysis sample (n = 78) compared to model testing sample (n = 19)

Calculate Outcome Measures

For the analysis of the simulation outcome measures from the model test sample were compared to output measures for the analysis sample. Table A8 provides the estimated power law distribution fit for online communities in the model testing sample.

Table 10. Model Test Subsample EstEstimated Power Law Fit	
Website	**Est. Power Law Fit (R^2)**
bid-alot.com/forum	0.92
bibleworks.com/forums	0.94
forums.builtbp.com	0.96
archive.burningsea.com/forums	0.93
cruisersforum.com/forums	0.97
developers.evrsoft.com/forum	0.97
forums.foxitsoftware.com	0.98
fsdeveloper.com	0.91
gamefileforums.com/forums	0.97
jpsoft.com/forums	0.95
forums.mxhub.com	0.96
npowersoftware.com/forums	0.90
programmersresource.com/forum	0.95
stormlabstuff.com/board	0.91
teamd86.com	0.87
bb.turtlesoft.com	0.98
vintage-computer.com/vcforum	0.97
vjforums.com	0.92
forums.winxpcentral.com	0.97
Average Power Law Fit (n = 19)	0.944
Standard Deviation	0.031

Appendix B

Measuring Power Law Distributions

A power law distribution is a long-tailed distribution of the form: $y = ax^b$. The *b* term is the power law distribution degree, a further measure of the shape of the power law distribution. A power law distribution's fit is a statistical estimate of how closely an observed distribution is to a power law distribution. To accomplish this, the computationally parsimonious method of calculating a linear regression fit of the log-log rank-frequency cumulative distribution function (Newman 2005) is used.

This linear regression-based method for measuring a power law distribution has the advantages of computational simplicity and of generating an estimated R^2 value (Clauset et al. 2009). Although this method has been criticized for failing to differentiate between multiple types of non-normal distributions and for potentially inaccurate estimations of power law distribution degree, it has been commonly used (Barabási and Albert 1999; Newman 2003) and is sufficient to identify which network mechanisms are consistent with the formation of the range of non-normal distributions observed in our online community reference sample. Caution is warranted, however, in making precise determinations based on the reported power law distribution degrees or in making differentiations between various extreme forms of non-normal distributions (e.g., exponential and power law).

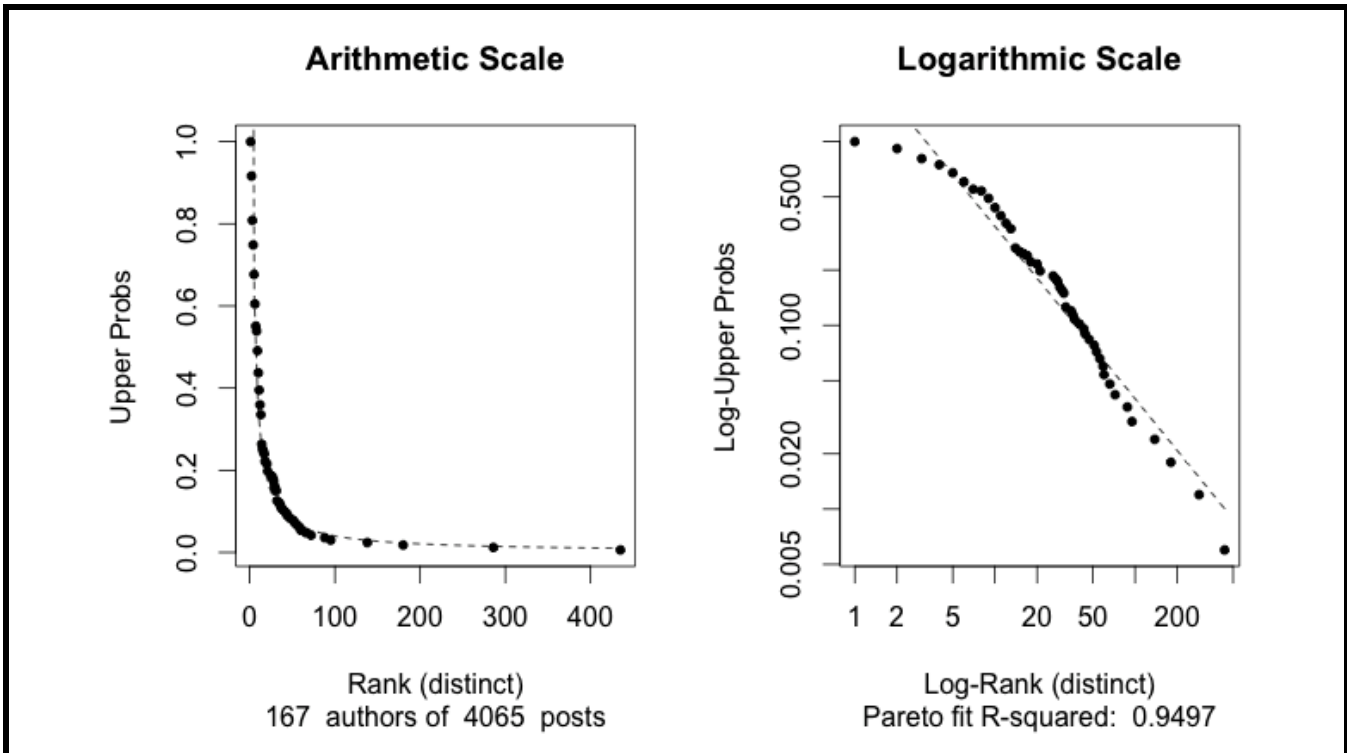


Figure B1. Fit of In-Degree Distribution of Online Community jpsoft.com/forums to Power Law Distribution

The detailed procedure used to calculate the online community participant in-degree rank/frequency Pareto (Type 1) distribution fit is

1. A full year of network history is gathered from an online community (e.g., the model calibration and model testing subsamples) or generated via simulation.

2. In-degree is calculated for each network participant as the total number of messages posted by other participants to message threads started by focal participant.

3. Participants are rank ordered from highest in-degree to lowest in-degree. Zero values are dropped.

4. The upper probability from the cumulative distribution function of distinct rank values is calculated for each distinct rank (shown in left side of Figure B1

5. Both distinct rank values (Y-axis in Figure B1) and upper probability values (X-axis) are log-transformed (right side of Figure B1).

6. The transformed values are fitted with an OLS regression. The adjusted R^2 value is the estimated power law distribution fit. The estimated slope coefficient is the estimated power law distribution degree (the *b* coefficient).

Steps 4, 5, and 6 are implemented in R using functions developed by (Clauset et al. 2009).

References

Andriani, P., and McKelvey, B. 2009. "From Gaussian to Paretian Thinking: Causes and Implications of Power Laws in Organizations," *Organization Science* (20:6), pp 1053-1071.

Barabási, A. L., and Albert, R. 1999. "Emergence of Scaling in Random Networks," *Science* (286), pp. 509-512.

Butler, B. S. 2001. "Membership Size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures," *Information Systems Research* (12:4). pp. 346-362.

Clauset, A., Shalizi, C. R., and Newman, M. E. J. 2009. "Power-Law Distributions in Empirical Data," *SIAM Review* (51:4), pp. 661-703.

Faraj, S., and Johnson, S. L. 2011. "Network Exchange Patterns in Online Communities," *Organization Science* (22), pp. 1464-1480.

Kolaczyk, E. D. 2009. *Statistical Analysis of Network Data: Methods and Models*, Heidelberg: Springer Verlag.

Law, A. M., and Kelton, W. D. 2000. *Simulation Modeling and Analysis*, New York: McGraw-Hill.

Matei, A., and Tillé, Y. 2006. "The R 'Sampling' Package," paper presented at the European Conference on Quality in Survey Statistics, Cardiff.

Newman, M. E. J. 2005. "Power Laws, Pareto Distributions and Zipf's Law," *Contemporary Physics* (46), pp. 7057-7062.

Preece, J. 2000. *Online Communities: Designing Usability, Supporting Sociability*, Chichester, UK: John Wiley & Sons.

R Development Core Team. 2009. "R: A Language and Environment for Statistical Computing," R Foundation for Statistical Computing, Vienna, Austria.

Ren, Y., Kraut, R., and Kiesler, S. 2007. "Applying Common Identity and Bond Theory to Design of Online Communities," *Organization Studies* (28:3), pp. 377-408.

Sanchez, S. M., and Lucas, T. W. 2002. "Exploring the World of Agent-Based Simulations: Simple Models, Complex Analyses," in Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers, December 8-11, San Diego, CA, pp. 116-126.