Beyond Friendship: Modeling User Activity Graphs on Social Network-Based Gifting Applications

Atif Nazir
Dept. of Computer Science
University of California–Davis
anazir@ucdavis.edu

Alex Waagen
Dept. of Applied Math
University of California–Davis
awaagen@math.ucdavis.edu

Vikram S. Vijayaraghavan
Dept. of Physics
University of California–Davis
vikram@ms.physics.ucdavis.edu

Chen-Nee Chuah
Dept. of Elecrical Engineering
University of California–Davis
chuah@ucdavis.edu

Raissa M. D'Souza
Dept. of Computer Science
University of California–Davis
raissa@cse.ucdavis.edu

Balachander Krishnamurthy
AT&T Labs–Research
bala@research.att.com

ABSTRACT

We employ user activity data from three highly popular gifting applications on Facebook to study the evolution of user activity on applications through the most commonly-used growth mechanism, namely Application Requests. We find user activity graphs differ from friendship graphs in large part due to the inherent directionality of user activity, and node transience. Our results show that, unlike degree distributions in friendship graphs, activity graphs exhibit strong asymmetry in in- and out-degree distributions, and that out-degrees are not accurately described by currently known parametric distributions. As such, user activity graphs cannot be simulated through existing intent- and feature-driven algorithms that can model friendship graphs.

We present a novel probabilistic growth model for user activity on the gifting genre of social applications. Our model decouples in- and out-degrees based on their distinct nature exhibited by our empirical data. We use the insight that regardless of increasing, declining or stable user activity, gifting application user activity exhibits the same graph structure. Our model produces synthetic graphs that consist of disconnected components with low clustering of nodes, and exhibit degree structures very similar to our real activity data. We discuss the benefits and shortfalls of our model and its applicability to other types of OSN-based applications, such as social games. To the best of our knowledge this study is the first to explore and model user activity growth processes on OSN-based applications.

Categories and Subject Descriptors: C.2.0 [Computer - Communication Networks]: General; H.4.3 [Information Systems Applications]: Communications Applications

General Terms: Measurement, Algorithms

IMC'12, November 14–16, 2012, Boston, Massachusetts, USA.

Author Keywords: Online Social Networks, Social Games, Social Gifting, Facebook, Applications, Algorithms

1. INTRODUCTION

Over the last five years, Facebook has become the most widely used Online Social Network (OSN) with a user base of over 950 million, and through its Developer Platform, Facebook supports a rich social application eco-system that has become a $6+ billion industry as of 2012. This growing importance of OSNs has spurred massive academic interest in OSNs, resulting in research that has focused on studying friendship graphs [15, 2, 23, 1]. Meanwhile, considerably less is known about *user activity* on social networks, even though researchers have argued the greater importance of studying user activity over simple friendship formation [22, 7, 29, 24, 25, 21]. This lack of research on user activity is due to privacy issues related to sharing of user activity data [12], which have resulted in lack of availability of such data for research purposes.

OSN friendship graphs differ significantly from user activity graphs (UAGs). We know that while friendship graphs consist of static friendship relations that rarely change between user pairs, UAGs are dynamic in nature due to the constant activity. Also, user activity on OSNs may or may not be reciprocal, whereas OSN friendships always are.[1] This means OSN friendship graphs are undirected, whereas UAGs are directed in nature. Furthermore, user activity on OSNs is not always based on the underlying friendship graph,[2] and UAGs exhibit temporal variance in graph properties that do not exist for OSN friendship graphs [16]. Due to these differences, existing work on friendship graphs is inapplicable to UAGs.

A number of questions regarding UAGs from social applications remain unanswered, including whether or not it is possible to model UAGs from social applications, just as researchers have modeled OSN friendship graphs. Our intent is to explore the possibility of modeling UAGs from all social applications on the largest OSN, Facebook. These

[1] Relationships on Twitter do not qualify as friendships, but rather as user-follower relations.

[2] For example, adaptations of board games such as Scrabble to Facebook applications have mostly been played with random application users rather than one's OSN friends.

applications utilize one or all four growth mechanisms available on Facebook to achieve growth in user base. These growth mechanisms are Application Requests (ARs), Newsfeed Stories, Emails, and Paid Advertising (see Section 3.1). Among these, only ARs are used by *all* Facebook applications. Based on this last observation, this paper focuses on understanding the growth of UAGs through ARs by studying, and modeling, UAGs from one genre of Facebook applications, namely, gifting applications. We say a social application is in the gifting genre if it consists of only one feature, and that this feature enables inter-user sharing of virtual gifts through ARs.

Our survey of the top 250 Facebook applications shows that gifting is the second-most popular genre of Facebook applications behind gaming, and that *all* Facebook gaming applications use gifting as a feature to increase growth. Furthermore, since gifting applications use only the AR mechanism, their UAGs are simpler to model than applications that use more than just the AR growth mechanism. *As a starting point to model UAGs from social applications, we investigate modeling UAGs from only gifting applications in this paper.* We use static snapshots of empirical UAGs from *three popular gifting applications* on Facebook to study the growth of UAGs. This approach of studying evolution of a graph by studying its static snapshots has been advocated previously ([10]), and it enables us to address the following questions regarding UAGs:

(1) How do UAGs from social applications differ from friendship graphs obtained from OSNs?

Unlike creation and deletion of friendships on OSNs, we have previously observed year-round dynamic patterns in the UAGs obtained from one gaming and three gifting applications [16]. In addition to daily and weekly variations in user activity, reliably high user activity is observed during special occasions and holidays (Valentine's Day, New Year, Christmas, etc.). Also unlike friendship graphs where friendships are almost always mutual, user activity is not always reciprocal. The lack of reciprocity in user activity gives rise to distinct in- and out-degree distributions. Furthermore, in our gifting UAGs, there is an explicit cost associated with initiation of activities (outgoing edges); i.e., the total number of activities that can be initiated by a user is capped at a limit. This is due to a throttling mechanism used by Facebook to limit spam in the network, which amplifies the distinction between in- and out-degree distributions.

(2) Existing graph models generate 'acceptable' synthetic friendship graphs [19, 13]. Do these models suffice when modeling user activity from social gifting applications?

We evaluate intent- and feature-driven algorithms (such as Nearest Neighbor and Forest Fire) reported to generate graphs that are statistically similar to OSN friendship graphs. We find the procedures employed in the original as well as existing hybrid versions of these algorithms are too simplistic to generate gifting UAGs. For instance, our results show that our gifting UAGs exhibit degree structures that are more complex than for friendship graphs. In particular, out-degree distributions in our UAGs cannot be explained simply by power-laws, and are not summarized by any known parameterized distribution. In contrast, OSN friendship graphs exhibit degree structures that are often explained by single power-laws [3].

(3) What key features should be considered to model UAGs from gifting applications? How do these features change with time?

Research on friendship graph modeling suggests a small number of metrics that are sufficient to summarize friendship graphs, i.e., clustering coefficients, degree distributions, and number/size of connected components [19]. We explore these metrics in conjunction with more focused user behavior metrics from our applications i.e., likelihoods of interacting with inactive users and targeting new users, and distributions of user lifetimes and number of daily interactions.

Our analysis of gifting UAGs shows that degree distributions as well as connected component size distributions tend to stabilize over approximately one week of user activity. Clustering coefficients are too small to reliably stabilize for our gifting applications, while the number of connected components do not stabilize, but instead may increase or decrease given the phase of the application's lifetime. We also observe that all metrics vary from week to week, depending on overall trends in application usage.

(4) Finally, is it possible to model user activity on a class of social applications that utilize the same underlying growth mechanism?

We use our understanding of the social application growth process to provide an algorithm tailored to a class of applications, namely *gifting*, that uses one particular growth mechanism, namely *ARs*. Unlike existing friendship models, our algorithm decouples in- and out-degrees to provide synthetic graph structures similar to our gifting UAGs. Our algorithm uses application-specific metrics such as active user duration, distribution of daily number of activities, etc. as inputs to facilitate accuracy of the synthetic graph produced. By varying inputs to this algorithm, we show its efficacy in generating *weekly* synthetic graphs for *all* of the gifting applications we study. We also perform rigorous analysis of the degree distributions produced by our algorithm. This analysis provides a mathematical formula that describes the expected distribution in activity level among users.

To the best of our knowledge, this is the first attempt to model UAGs from OSN-based applications. Due to similarity of structure and the pervasive use of ARs, we believe our model is applicable to other Facebook gifting applications as well. Our findings can be leveraged to forecast a gifting application's usage. The synthetic UAGs generated by our model provide an alternative to sharing large data sets that may violate user privacy.

We begin this paper with a discussion of related work in Section 2, followed by a discussion of our methodology, including the graph metrics we use to model gifting UAGs, in Section 3. We analyze measurements for our selected graph metrics in Section 4, and present our UAG growth model in Section 5 with a discussion of its potential uses. We discuss future work and conclude the paper in Section 6.

2. RELATED WORK

Spurred by the increasing relevance of OSNs to online user activity, there has been an increase in analyses of OSNs over the last few years [11]. While some researchers have analyzed usage patterns of popular OSNs, others have focused on graph theoretic properties of OSNs such as YouTube [15].

Research into OSN network growth has resulted in multiple algorithms that produce synthetic graphs to mimic OSN friendship graph properties. These algorithms derive concepts from earlier graph theory [18], including algorithms for offline social network growth, such as Nearest Neighbor, Random Walk, Barabasi-Albert, and so on [19]. Moreover, some research marries preferential attachment with latent node characteristics to study OSN network growth [9].

Researchers have also studied time-varying OSN graphs [20, 8]. Gummadi *et al.* report on the strength of user interactions on Facebook using indigenous OSN applications in [25]. They find that communication decreases among users from the point of friendship formation due to relationship fatigue. Leskovec *et al.* propose that graphs over time do not follow the same patterns or exhibit the same attributes as they did when they were initially formed, and presents the Forest Fire algorithm to model OSN network growth [13]. Other research has focused on information dissemination patterns on OSNs [21, 2]. Li *et al.* have suggested that in complex networks, relying on statistical methods cannot provide an idea of which features are essential to the topology's formation [14]. Instead, they propose that knowledge of the building blocks of network activity is essential to understand growth of networks. Furthermore, Holme *et al.* propose studying static snapshots of graphs to study evolution of temporal networks [10].

The lack of availability of real data sets from OSNs has slowed research into growth patterns on OSNs. This is especially true for social applications on OSNs, for which high-level statistical analyses exist ([5, 16, 17, 6]), yet whose growth processes have not been studied.

Our social applications provide us a unique opportunity to study the growth of OSN-based applications [16, 17]. We believe the observation of Gummadi *et al.* on user fatigue holds true for social applications as well, since application novelty can explain early application growth and user fatigue could account for its decline. We find that growing and aged graphs exhibit different patterns ([13]) for UAGs as well. To the best of our knowledge, this paper is the first to study and model growth of user activity on social applications.

3. METHODOLOGY AND RATIONALE

Our end-goal is to study growth processes of all OSN-based applications. However, due to limited access to OSN application data, we use only applications on the largest OSN, Facebook, as a case study.

3.1 Growth Mechanisms of OSN Applications

Facebook applications allow subscribed users to interact with non-subscribing users to encourage application growth on the OSN. We define a *subscribing* user on a Facebook application as someone who has 'logged in' to the application through an installation screen presented by Facebook for that application. Moreover, online social applications belong to different genres. While some of the more complex applications (games) target audience that seek high engagement in OSN applications,[3] other simpler applications (gifting) target casual users i.e., users that spend only a short time on social applications.

Application statistics websites ([27], [26]) rank Facebook applications according to Monthly Active Users (MAU) [16]. Our review of these rankings indicates that gaming and gifting applications are the two largest sets of Facebook applications. In particular, our review (using Developer Analytics [27]) in January 2012 showed that 74.4% of the top 250 Facebook applications by MAU were Flash or text-based gaming applications, while 9.43% were gifting applications, followed by other smaller genres of applications.[4]

We study growth processes for social applications using the OSN-based application architecture (Figure 1(a)), discussed in [16]. These growth processes depend on the growth mechanisms employed by social applications. Facebook provides the following growth mechanisms:

- **Application Requests (ARs)**: Facebook users are able to send ARs to their friends, allowing them to recruit friends to applications free of cost. The AR growth mechanism is illustrated by steps 1 and 5 in Figure 1(a). If User A sends an AR to User B, the sending user performs step 1 and the OSN (asynchronously) performs step 5 for User B. When a user receives an AR, she can either *accept* it or *ignore* it. Accepting an AR presents non-subscribing users with an installation page, and subscribed users are shown the reason(s) they were sent an AR. An AR is sent by a subscribing user to any friend on Facebook, whether subscribed to the application or not. Facebook limits the number of these outgoing requests on a given application through an internal spam-control algorithm. For our applications, the daily per-user AR limit for an application was typically 20, but could be as high as 60. There are, however, no limits on ARs a user may receive.
- **Newsfeed Stories**: Newsfeed stories are personalized posts on a user's profile made through applications by subscribed users. Facebook controls the visibility of Newsfeed stories to users' friends through an internal algorithm that prioritizes a story's importance to the viewing user. A Newsfeed story can be posted by a subscriber to any friend, whether subscribed to the application or not. Our review shows 88% of the top 250 applications prompt users to share a Newsfeed story in at least one sequence of actions. Note that due to the algorithm Facebook uses to display Newsfeed stories to users, as well as the passive nature of Newsfeed stories compared to ARs, the latter have been more reliable for application growth in our experience.
- **Emails**: Pending a subscribed user's approval, Facebook allows applications to send e-mail notifications to their personal e-mail addresses when an event occurs. Since users often perceive allowing e-mail access to social applications as a security risk, developers do not use this communication channel often. We found only 36% of the top 250 applications at least ask users for their email addresses, but it is not practical to measure how often this mechanism is utilized due to the complex nature of user flows in most gaming applications.
- **Paid Advertising**: Application developers on Facebook may purchase advertisements to boost subscription growth. However, we are unable to discern how widely this mechanism is used by analyzing the applications alone. Anecdotally, most social games tend to use advertisements to increase subscription growth.

Our review of the top 250 Facebook applications suggests applications may use one or all growth mechanisms.

[3] Higher user engagement translates into higher average revenue generated per subscriber.

[4] Since the genre listings of applications can be inaccurate, we categorized the top 250 applications manually.

However, *all* applications use ARs. *Our data allows us to study specifically the growth process arising from use of ARs through UAGs from Facebook-based gifting applications.* We define a *gifting application* as an application that 1) uses only ARs for growth and inter-user communication, and 2) only allows users to share themed images (gifts) through ARs with their Facebook friends.

3.2 Measuring Gifting Application UAGs

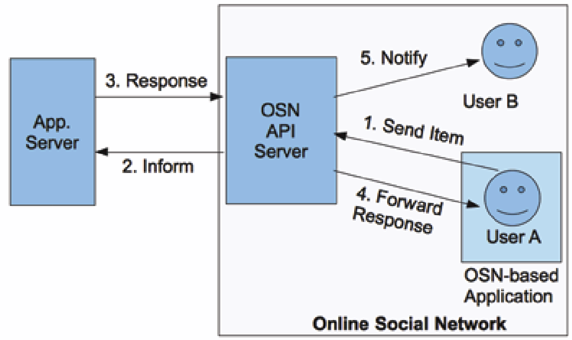
We employ UAGs from three Facebook gifting applications to study application growth through the AR growth mechanism. The architecture of the Platform used is shown in Figure 1(a). We say a single user *activity* occurs on a gifting application when steps 1 and 5 are performed by the sending user A, and the OSN, respectively. The third-party application servers record the sending and receiving users' Facebook UIDs with the time at which the activity was generated. We only studied anonymized data for our research.

Each node in the UAG represents a Facebook user, and each directed edge from User A to User B represents an AR sent from User A to User B. Note that our UAGs are multigraphs, i.e., User A may have multiple directed edges to User B. Our UAGs were obtained through server-side measurements of the following Facebook-based gifting applications, which were owned and operated by Manakki, LLC:

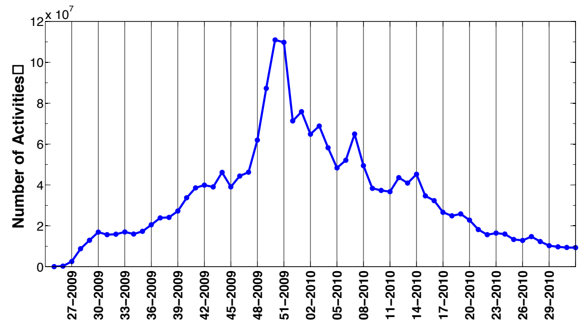
- **iHeart**: Launched in June 2009, iHeart grew to 30 million monthly active users by December 2009, and was installed by 76 million users on Facebook by August 2010. Our user activity traces capture 2.2 billion activities. Users share decorative heart-shaped graphics with their Facebook friends.
- **iSmile**: Launched in August 2008, iSmile was installed by 43 million users on Facebook by August 2010. Our user activity traces for iSmile capture 1.5 billion user activities. Users share graphics of smiling characters with their Facebook friends.
- **Hugged**: Launched in February 2008, Hugged was installed by 28 million users on Facebook by August 2010. Our user activity traces for Hugged capture 1.6 billion user activities. Users share graphics of hugging characters with their Facebook friends.

iHeart was a very popular application that ranked in the top three applications on Facebook by MAU in December 2009. We use iHeart to drive and evaluate the development of our UAG growth model, and employ UAGs from Hugged and iSmile to test this model's applicability to other gifting applications. Figure 1(b) shows user activity on iHeart for 13 months after its launch, and indicates three distinct periods of activity on iHeart: the ramp-up user activity period (until week 49, 2009), the short-lived peak user activity period (weeks 50-51, 2009), and the declining user activity period. The sharp fluctuations in user activity in all three phases indicate real-world events/holidays, where user activity usually spikes before the occasion and then falls during the event, with exceptions such as Mother's Day (week 19, 2010). This three-phase lifetime user activity is evident on all of our social applications, is attributed to application novelty and eventual user fatigue, and is less prominent on longer-lasting applications such as Farmville [28].

In order to perform analysis that is representative of all three phases of our applications' user activity and lower processing times for our analysis, we use the periods of activity for iHeart shown in Table 1. This sample of user activity periods captures all variations in application lifetime phases, stable/increasing/declining user activity and special real-world events. We use the same 10 weeks' data from Hugged and iSmile to test our UAG growth model, but only list week 34-2009 in Table 1 for brevity.



(a) The Facebook Platform architecture.



(b) The total number of hearts sent per week on iHeart across 13 months. We see three user activity periods for iHeart: increasing activity (upto 50-2009), peak activity (50-2009) and declining activity (01-2010 onwards). Sharp dips/spikes in user activity are due to holidays and special real-world events. From left to right, these dips are due to: Halloween, Thanksgiving, Christmas/New Year, and Valentine's Day.

Figure 1: The Platform architecture used by our applications, and user activity for iHeart.

In our traces, a week's user activity starts at 12:00:01AM Pacific Time on Sunday, and ends on the following Sunday at 12:00:00AM. We discuss the reason for selecting weekly user activity periods over shorter and longer choices in Section 4. We performed analysis and simulations for all listed weekly activity periods. However, due to the similarity in results from a given phase's weekly periods, we only present results for one week from each phase: we use week 34-2009 as a representative week for the pre-peak phase, week 50-2009 for the peak phase, and week 26-2010 for the post-peak phase.

We discuss the metrics we use to model our UAGs next.

3.3 Understanding Social Application Growth

Existing friendship models generate synthetic graphs that reproduce the following key graph properties:

- *Clustering Coefficient (CC)*: The CC of a node $v \in V$ is the ratio of number of edges between neighbors x of v (such that $\exists(x, v) \in E$) and the total number of

Table 1: Relevant measurements for weekly UAGs from iHeart, Hugged and iSmile. Our weekly activity periods belong to all three phases of an application's lifetime: pre-peak (P-), peak (P) and post-peak (P+). Section 3.3 defines α, β, γ^d, γ_m, M and N_0.

App.	Week	Phase	Users	Activities	α	β	γ_d	γ_m	M	N_0
iHeart	34-2009	P-, Stable	9.85M	11.8M	0.541	0.875	1.75	0.77	20	0.52M
	38-2009	P-, Increasing	14.7M	18.4M	1.82	0.74	0.512	0.867	20	0.80M
	45-2009	P-, Spiky	22.3M	29.6M	0.477	0.863	1.71	0.80	20	1.17M
	50-2009	P, Stable	44.7M	110M	0.414	0.899	1.75	1.09	60	1.37M
	01-2010	P+, Declining	31.4M	43.8M	0.459	0.879	1.72	0.87	24	1.57M
	06-2010	P+, Spiky	30.95M	47.59M	0.452	0.869	1.57	0.85	24	1.61M
	14-2010	P+, Declining	19.0M	23.7M	0.516	0.878	1.80	1.09	20, 24	1.12M
	19-2010	P+, Spiky	11.59M	13.0M	0.592	0.889	2.14	0.99	26	0.83M
	26-2010	P+, Stable	9.89M	11.1M	0.629	0.904	1.95	0.93	26	0.62M
	30-2010	P+, Declining	6.51M	6.86M	0.644	0.899	2.05	1.07	20, 26	0.52M
iSmile	34-2009	P-, Stable	5.28M	7.17M	0.697	0.901	2.67	0.69	20	0.28M
Hugged	34-2009	P-, Stable	2.72M	3.55M	0.727	0.899	2.79	1.06	20	0.14M

edges possible between those neighbors. The CC of a graph is the average of individual nodes' CCs.

- *Degree Distribution*: The degree distribution $\forall k : P(k)$ of a graph is the fraction of nodes with degree k. If there are N nodes where n_k have degree k, $P(k) = n_k/N$. We distinguish between in- and out-degrees.
- *Connected Components*: Two nodes x and y belong to one connected component if an undirected path between x and y exists. A component's nodes are only connected with other nodes in the same component. We look at *number* of components, as well as *percentage of users in the largest* components.

Existing research uses these properties to summarize the static structure of OSN graphs [19]. We want a growth model that reproduces the above properties, including distinct in- and out-degree distributions, for our UAGs. Our findings show these graph metrics stabilize around a week's aggregate user activity for our applications (Section 4). We use this finding to simulate only weekly user activity to simplify our growth model. Our model differentiates between active, and inactive users. *Active* users send at least one AR in a week's UAG, and *inactive* users only receive, and do not send, ARs in a week's UAG.

The social application growth processes are dependent on use of different growth mechanisms. We believe the following list of application-specific parameters must be used in a growth model for gifting applications.

- *Probability of Sending to New Targets (α)*: This is the probability a user will target a previously unseen user through ARs in our UAGs.
- *Probability a User Remains Inactive (β)*: Inactive users may or may not be subscribed to our applications. The subscribed inactive users, even though they may visit the application, do not send ARs. The parameter β represents the probability a user will remain inactive in a weekly UAG.
- *Active User Duration (d_x)*: Nodes in UAGs are more transient in nature as compared to friendship graphs. In order to capture this transience in our model, we use the distribution of the number of days an active user appears in a given week's UAG as a parameter. Our measurements suggest d_x is power-law distributed, with exponent γ_d.
- *Daily ARs Sent (m_x)*: Even though our UAGs show limits on ARs per day, users may not exhaust these limits. Furthermore, some users may visit an application but not send any ARs at all. This parameter captures the distribution of number of ARs users send on a given day, which is important for generating accurate degree distributions. Our measurements suggest m_x is power-law distributed, with exponent γ_m.

Note that user activity on social applications is continuous, i.e., users in one time period are responsible for incoming users in the consecutive time period. We have previously reported that almost 80% of all AR acceptees visit the application within 48 hours of receiving the AR [16]. If we consider weekly snapshots of user activity, a proportion of users never receive ARs in the current week, yet appear in the UAG regardless. These are the *seeding users* responsible for continued activity on the application in the given week. Along with the daily AR limit (M), we use the number of seeding users (N_0) as an *external constraint* in our model.

4. CASE STUDY: UAGS FROM IHEART

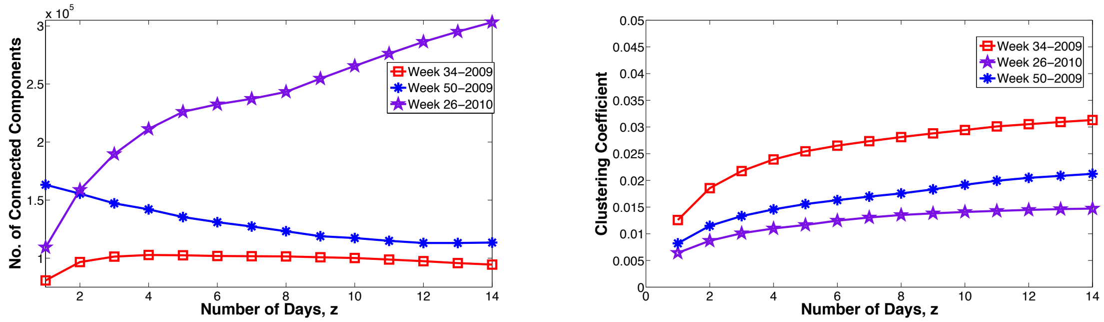
This section presents the defining features of UAGs observed in our most popular application, iHeart. As mentioned in Section 3.1, iHeart shares structure and design with other gifting applications on Facebook. We build our algorithm to simulate iHeart's UAGs first, and then test it on our other gifting applications. We use the activity periods listed in Table 1 for this analysis.

Our findings suggest graph metrics do not stabilize in hourly or daily activity traces. We measure convergence times for the key graph metrics to gauge the minimum period required for metric stability in our UAGs.

4.1 Convergence of Key Graph Metrics

We define the *convergence* time Z as the amount of time it takes for metric values to stabilize for our UAGs. More specifically, $Z = \min z$ such that $f(z+1) - f(z) \approx 0$, where $f(z)$ represents metric values measured from a UAG of the first z days in a given time period.

Degree Distributions: We measured Z for in- and out-degree distributions on iHeart by comparing aggregated user activity (starting at Sunday) for the 10 time periods listed in Table 1. The function $f(z)$ represents the CDF of user activity over z days for the in- or out-degree distribution.



(a) Number of connected components. (b) Clustering coefficient.

Figure 2: The number of connected components and clustering coefficients over 14 days of aggregated user activity for iHeart, starting at Sunday for weeks 34-2009 (pre-peak), 50-2009 (peak) and 26-2010 (post-peak).

The resulting $f(z+1) - f(z)$ values, or Euclidean Distances (EDs) indicate a small distance between $f(2)$ and $f(1)$ (i.e., one and two days' aggregate user activity) for both in- and out-degrees, and that this difference decreases monotonically as z increases. The in- and out-degree distributions stabilize roughly at $z = 7$ days, with ED $\leq 2.5\%$ for all 10 weekly activity periods. An exception occurs when new content is introduced during week days, where out-degree distribution EDs spike to up to 34%. However, we observe ED $\leq 2.5\%$ as we approach the end of the week.

Connected Components: We look at two metrics with regards to connected components in our UAGs: the *number* of connected components and the *percentage of users in the largest* connected component.

In general, we find that as more users interact through ARs, the *number* of distinct connected components grows. However, unlike degree distributions, in our applications the number of connected components behaves differently for pre-peak, peak and post-peak activity periods. In the peak activity period the number stabilizes over the course of a week, but it does not converge in either pre-peak or post-peak activity periods. Figure 2(a) shows this behavior for the representative periods on iHeart. More specifically, the number of connected components declines when z increases for pre-peak activity periods, indicating that increasing AR activity between users merges the largest component with smaller, fragmented components. In post-peak activity periods, in contrast, the number of connected components increases as z increases. User fatigue is a possible explanation for this phenomenon, i.e., users take longer to return to the application leading to formation of small (up to 110 node) components that do not connect with the largest component as quickly as in the pre-peak periods.

The percentage of users in the largest component, however, does stabilize for our UAGs. If $f(z)$ represents the percentage of users in the largest connected component, we find that $f(z+1) - f(z) \leq 0.5\%$ where $z \geq 6$ for all time periods considered (Table 1). For different time periods, however, $f(7)$ varies between 90.17% and 94.89%. Note that the second largest components for our weekly UAGs are of negligible size (less than 20 nodes).

Clustering Coefficients (CC): Figure 2(b) shows the CC for undirected iHeart UAGs for three representative periods 14 days of activity. We have previously discussed the effect of user engagement on CCs in UAGs from social applications in [16], where we showed a highly engaging gaming application exhibited high node clustering (CC = 0.8). In contrast, Figure 2(b) shows the CC is very small for iHeart, but that increasing z results in slightly higher CC. Note that the aim of this study is to provide a growth model that approximates gifting UAGs as a first step. Therefore, we seek to generate synthetic graphs with low or no clustering of nodes rather than graphs with precisely the same number of clusters as in our empirical UAGs.

A weekly snapshot of user activity captures a full cycle of low and high user activity for OSN-based applications [16]. Moreover, since gifting applications are non-recurrent (compared to social gaming applications), less than 10% of users return in a consecutive week. This high churn of users means every week's user activity can almost be treated independently. Our results show both degree distributions and percentage of users in the largest connected components stabilize within 7 days of user activity. Therefore, we focus only on weekly time periods for our first UAG growth model.

4.2 Approximating Degree Distributions

Existing research indicates power laws may partially explain degree distributions in OSN graphs [15, 19]. However, our measurements indicate that known parametrized distributions do not accurately fit the full range of the degree distributions for gifting UAGs. Unfortunately, this means that the simple growth mechanisms known to give rise to the parameterized distributions (such as preferential attachment which leads to power laws, or multiplication of independent random variables with leads to log-normal distributions) do not suffice to explain growth of our UAGs. To demonstrate the lack of fit, we attempt to fit the candidate distributions shown in Table 2 to our real user activity data.

We perform Goodness of Fit (GoF) tests for potential candidate distributions (Table 2) for iHeart UAGs. We use the Kolmogorov-Smirnov (KS) statistic between the data and parametric fits to measure applicability of the distributions to our UAGs. The KS statistic is suggested as a reliable GoF measure for heavy-tailed distribution approximation [4]. We do an exhaustive search of the parameter space to find the best fit that minimizes the error for a given distribution.

First, consider *in-degree distributions*. The KS statistic indicates that both PL and PLED provide fits with good statistical accuracy, while LN, SE and EXP are orders of magnitude higher for all weekly UAGs (see Table 3). Be-

Distribution	P(x)	Valid Range of Parameters
Power law (PL) or Algebraic Decay	$ax^{-\gamma}$	$0 < a, \gamma < \infty$
Power law w/ Exp. Decay (PLED)	$ax^{-\gamma}\exp(-x/\tau)$	$0 < a, \gamma, \tau < \infty$
Exponential (EXP)	$a\exp(-x/\tau)$	$0 < a, \tau < \infty$
Stretched Exp. (SE)	$\frac{a}{b}(x/b)^{a-1}\exp(-(x/b)^a)$	$0 < a, \tau < \infty$
Log-normal (LN)	$\frac{1}{x\sqrt{2\pi\sigma^2}}\exp(-(\log x-\mu)^2/2\sigma^2)$	$0 < \mu, \sigma < \infty$

Table 2: Candidate distributions to approximate our UAGs' degree structures.

Week	GoF	In-degree				
		PL	PL-EXP	SE	LN	EXP
34-2009	KS	1.16E-04	3.21E-05	0.1412	0.0806	0.0694
50-2009	KS	7.87E-04	2.85E-04	0.0695	0.1361	0.0548
26-2010	KS	3.49E-04	1.28E-04	0.1052	0.0623	0.0536

Table 3: GoF measures for in-degree distributions for weekly iHeart user activity.

Week	GoF	Out-degree, Bulk				
		PL	SE	LN	EXP	PLED
34-2009	KS	3.99E-03	0.0914	0.0744	0.1857	9.69E-03
50-2009	KS	0.0163	0.3687	0.2886	0.4533	0.0571
26-2010	KS	5.04E-03	0.2375	0.1654	0.2886	0.0834

Table 4: GoF measures for out-degree (bulk) distributions for weekly iHeart user activity.

Week	GoF	Out-degree, Tail				
		LN	PL	EXP	SE	PLED
34-2009	KS	5.39E-03	6.58E-03	5.77E-03	5.41E-03	5.49E-03
50-2009	KS	3.82E-04	3.34E-04	4.49E-04	4.38E-04	6.289E-03
26-2010	KS	7.32E-03	7.11E-03	7.75E-03	7.74E-03	0.0195

Table 5: GoF measures for out-degree (tail) distributions for weekly iHeart user activity.

Week	In-degree Fit	Out-degree Fit		
	γ_i	γ_o	μ	σ
34-2009	3.394	0.7443	1.661	1.135
50-2009	3.628	0.7436	2.145	1.296
26-2010	3.457	0.9202	1.9414	0.9143

Table 6: Parameters of the distributions that best fit the data. In-degrees were fit using PL with exponent γ_i, while out-degrees' *bulk* were fit using algebraic decay with exponent γ_o and *tail* with LN with mean μ and standard deviation σ.

tween PL and PLED, the latter is more accurate for the early portion of the distribution, yet PL captures tail events better, is a simpler distribution, and achieves overall accuracy very close to the PLED fit. Figure 3 shows an example PL fit (starting at $k = 12$) to the week 34-2009 UAG from iHeart, and Table 6 shows the best-fit distribution parameter values for the three representative weeks.

Now consider *out-degree distributions*. Figure 3 shows an example out-degree distribution for week 34-2009. The algebraic 'steps' in weekly out-degree distributions occur at multiples of daily AR limits (shown as M in Table 1). Note that more than 88% of AR senders in any week have out-degrees less than the daily AR limit, i.e., only a small fraction of users send 20 or more ARs in a given week.

We find that the out-degree distributions are best fit by splitting the data into two parts at the daily AR limit: we call the data up to the daily AR limit the *bulk* of the distribution, while degrees above the daily AR limit are the *tail* of the distribution. This is because the behavior exhibited by the *bulk* differs significantly from the *tail*, highlighting the fact that 1) most users do not exhaust the daily AR limit, and 2) most users visit gifting applications once a week due to low engagement [16]. The KS statistic indicates the out-degree distributions for the *bulk* are explained best by an algebraic decay (PL), followed by PLED, SE, EXP and LN, in that order, whereas the *tail* can be reasonably fit using either an LN or PL distribution, as shown in Tables 4 and 5. The KS statistic is not extremely sensitive to tail events, and visual inspection confirms that LN provides a better fit to the tail for all weeks studied. Figure 3 shows an example best fit for out-degrees, and Table 6 shows the best-fit distribution parameter values for the three representative weeks. We performed these fits by removing all multiples of daily AR limits to reduce distortion in KS values.

Thus a power law can reasonably fit the in-degree distribution, but no simple parametric distribution describes the full out-degree distribution. Even when fitting the out-degree with two different distributions, the values of the parameters for these distributions vary considerably as seen in Table 6. As such, we cannot explain UAG growth via a simple mechanism like preferential attachment or multiplicative growth. Our findings are valid for all other weekly periods (Table 1) as well.

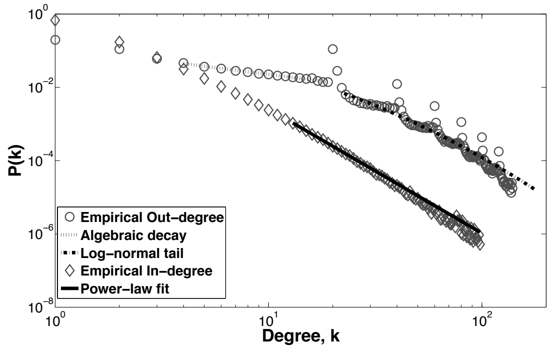


Figure 3: In- and out-degrees for weeks 34-2009 along with parametric distributions that fit. In-degrees were fit using PL (exponent 2.90), while out-degrees' *bulk* were fit using algebraic decay (exponent 0.744) and *tail* with LN (mean $\mu = 1.66$ and standard deviation $\sigma = 1.135$).

4.3 Relevant Metrics for Modeling Growth

As discussed in Section 3.3, certain key metrics are needed to model gifting UAGs. These metrics will serve as input parameters for our modeling algorithm.

Probability of Sending to New Targets (α): Table 1 shows the probability of sending hearts to a new user ranges from 0.41 to 0.73 in weekly UAGs. Also, users are more likely to target new users in lower activity periods.

Probability a User Remains Inactive (β): Our measurements show (Table 1) the probability a randomly selected user is inactive throughout a week is in a very narrow range between 0.85 and 0.91 for iHeart, i.e., it is more or less stable throughout iHeart's lifetime.

Active User Duration (d_x): Our measurements show between 70% to 75% of active users in a given week are only active for a day, and this percentage decays approximately as a power law with number of days active (e.g., only between 1.5% and 2% users are active for all seven days). Table 1 shows the weekly PL exponent of this distribution (γ_d).

Daily ARs Sent (m_x): Our measurements show that the distribution of ARs generated per day follows a power law for iHeart. Our algorithm uses the exponent, γ_m, of this PL distribution, and its values are shown in Table 1.

Additionally, Table 1 shows measurements for our *external constraint* on the seeding users (N_0) for a given week. We observe the lowest proportion of seeding users at peak activity, and highest proportion during lowest user activity. Week 45-2009 (Thanksgiving) is an exception since only four days of the week see high user activity, followed by three days of very low user activity (Figure 1(b)) as the Thanksgiving weekend kicked in.

4.4 Fitting Graph Models to iHeart UAGs

Existing research on growth processes on OSNs has yielded viable candidate models for friendship graphs [19]. We tested some of the more promising algorithms for our applications, in particular the original versions of the Nearest Neighbor, Watts-Strogatz, Barabasi-Albert and Forest Fire algorithms. All of these algorithms rely mainly on preferential attachment, and do not distinguish between behaviors of different types of nodes resulting in out-degree structures that are not representative of our UAGs. However, these algorithms do provide decent approximations of in-degrees for our UAGs. This is evident from the KS statistic values measured using synthetic graphs from these models against all weekly degree distributions from iHeart. Example KS statistic values for the Nearest Neighbor and Forest Fire algorithms against the week 34-2009 UAG from iHeart are shown in Table 7.[5] Figure 4 shows the degree structures that resulted in the lowest average KS values from these models against those from the week 34-2009 UAG for iHeart.

Period	Model	KS_{in}	L_{in}	KS_{out}	L_{out}
34-2009	FF	0.2117	7.81	0.6671	70.3
34-2009	NN	0.1768	7.65	0.7839	183.01

Table 7: Error measurements for in- and out-degrees for the weekly iHeart data using directed Forest Fire (FF) and Nearest Neighbor (NN) models.

5. MODELING GIFTING APPLICATIONS

The algorithm described in this section produces synthetic graphs with degree structures similar to UAGs for our gifting applications while preserving low clustering of nodes. We will show how our algorithm succeeds in capturing the difference in in- and out-degree distributions, which were not reproduced by previous models. We also provide steady-state equations (Section 5.4) for the synthetic distributions as a means to gauge user activity levels on gifting applications without running possibly time-consuming simulations.

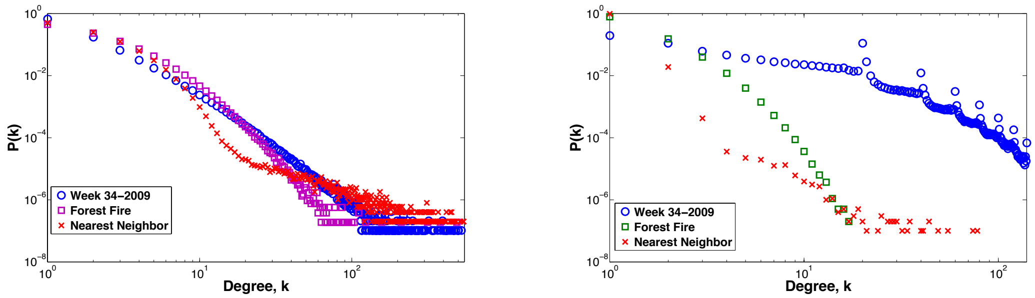
5.1 Our UAG Growth Model

We have shown our UAGs exhibit similar structure across different weekly periods in terms of low clustering of nodes, and class of distribution functions that approximate the real degree distributions. However, the parameter values for these distributions that best fit the data differ significantly from week to week. This variability in UAGs must be captured by our algorithmic model. The basics of our growth model are as follows:

Our UAG growth model operates at discrete *timesteps* t, where each timestep represents a day of user activity. Let G_t denote the graph at the beginning of timestep t. The graph G_0 is a set of N_0 isolated nodes, which correspond to initial seeding users. Seeding users send ARs in each turn they are active. They send ARs to new users (i.e., 'recruit' them) with probability α, and to existing users chosen preferentially by in-degrees otherwise. When a user is recruited, with probability β she will choose to remain inactive throughout the process, and otherwise will be active at the beginning of the next timestep. Each user is assigned a fixed duration and daily output of ARs. The duration d_x of user x corresponds to the number of timesteps x is active, and the per-timestep output m_x of x corresponds to the number of ARs sent by x daily. After 7 timesteps, the resulting synthetic graph G_7 is produced. The structure of G_7 should be comparable to our UAGs.

Algorithm 1 shows the pseudocode for our UAG growth model. This algorithm uses *four* input parameters (α, β, γ_d and γ_m as described in Section 3.3), as well as two external constraints M and N_0. In the pseudocode, IPL(γ,t) and

[5]The L metric is defined in Section 5.2.



(a) In-degree distribution comparison.

(b) Out-degree distribution comparison.

Figure 4: Visual comparison of degree structures for Nearest Neighbor, Forest Fire algorithms with the week 34-2009 UAG from iHeart.

RPL(γ,t) are integer- and real-valued PL distributed random variables (respectively) of exponent γ with a cut-off at t. Moreover, the variable act_x is the number of timesteps a user has previously been active, and $diff_x$ is the difference between $\lfloor m_x \rfloor d_x$ and the ultimate out-degree $\lfloor m_y d_y \rfloor$ of x.

5.2 Synthetic vs. Empirical UAGs

We implemented Algorithm 1 in Python and tested it on all 10 weeks of iHeart, iSmile and Hugged user activity. However, for brevity's sake, we only present results of simulations using UAGs from weeks 34-2009, 50-2009, 26-2010 for iHeart, and week 34-2009 for iSmile and Hugged.

We evaluate our simulated graphs' degree structures using two goodness of fit measures. In Section 4.2 we used the KS statistic to evaluate our degree distribution fits. However, the KS metric does not capture variance in tails as accurately as, say, the chi-squared metric. Therefore, we use an additional quantity we call L, which is analogous to chi-square on a log scale. Given two positive, integer-valued random variables X and Y, we define:

$$L(X,Y) = \sum_{k=1}^{\infty} [\log \Pr(X=k) - \log \Pr(Y=k)]^2 \log(\frac{k+1}{k})$$

Note that $\log(\frac{k+1}{k}) = \log(k+1) - \log(k)$. Hence L can be viewed as a discrete equivalent of the L_2 measure $d(f,g) = \int((f(x) - g(x))^2 dx)$ on a log-log scale.

Table 8 shows the KS and L values for our simulations using the 4-parameter model that is Algorithm 1, and Figure 5 shows the plots for the same. The KS and L values suggest the empirical degree structure is reasonably well-modeled in the 'typical' week represented by week 34-2009. For comparison, note the significantly larger L_{out} values in Table 7 for out-degree distributions generated by the Forest Fire and Nearest Neighbor models for week 34-2009 on iHeart. The larger L_{out} values demonstrate that Algorithm 1 performs significantly better than either Forest Fire or Nearest Neighbor in modeling out-degree distributions for our applications. Furthermore, L_{in} values in Tables 8 and 7 show that Algorithm 1 significantly outperforms both Forest Fire and Nearest Neighbor algorithms for in-degrees as well. Given the simplicity of the attachment rule used in our model, the fact that the in-degree distributions are captured so well is surprising. This is evidence that the dynamics of the real system may be similar to the 'Preferential Attachment with recruitment' dynamics in our model.

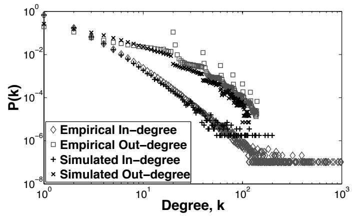
Our simulations using the 4-parameter model, however, result in synthetic graphs with comparatively large L values for the peak (week 50-2009) and post-peak (week 26-2010) periods for iHeart. This is a result of our inability to measure the degree of preference a user employs when sending ARs. Occurrences such as this are due to the inability of simple metrics to capture behavior of users in social graphs [19]. To remedy this, we introduce two additional parameters and refer to this as the *6-parameter version* of the model shown by Algorithm 1. The 6-parameter model is identical to the 4-parameter version in all aspects except the choice of the Active User Duration d_x and Daily ARs Sent m_x. In the 6-parameter model, these are approximated by LN distributions rather than PL distributions. LN distributions require *two* parameters whereas PL distributions only required one, so that the total number of parameters becomes 6.

Table 8 shows the L and KS values for our UAGs using the 6-parameter model. Our 6-parameter model simulations show that in some cases the error measurements actually grow larger compared to our 4-parameter model, but only slightly so. On the other hand, some of the error in the measurements can be dramatically reduced (out-degrees for iHeart). In week 50-2009, L_{out} drops from 18.08 to 0.62 when the 6-parameter model is used, and in week 26-2010 it drops from 7.75 to 1.44. An example resulting distribution using the 6-parameter model is shown in Figure 6.

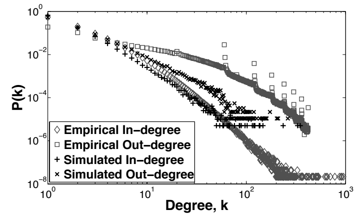
Furthermore, as mentioned in Section 4.1, our UAGs exhibit very low clustering of nodes. Our UAG growth model's synthetic graphs similarly exhibit little or no clustering of nodes with CCs less than 0.0005. Our synthetic graphs also consist of connected components of size distributions similar to our UAGs. That is, more than 90% of users belong to one connected component, and the second largest component is of negligible size. We provide proof of the disconnectedness of our model's synthetic UAGs in Section 5.4.

5.3 Observations for our Growth Model

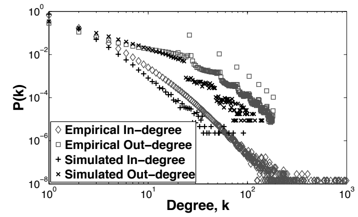
By demonstrating our algorithm works well on three gifting applications, we have shown it is possible to model UAGs for a class of social applications. *By decoupling in- and out-degrees and recognizing different types of users, our model provides a very close approximation of degree structures that result from use of the AR growth mechanism.* As in our empirical UAGs our model produces disconnected graphs such that most nodes belong to one connected component. Our synthetic graphs also exhibit little to no clustering of nodes.



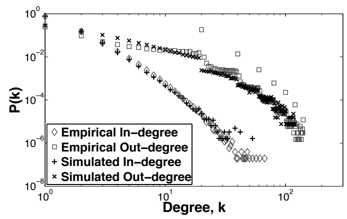
(a) Week 34-2009 for iHeart.



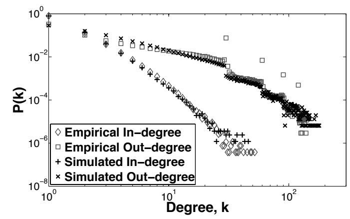
(b) Week 50-2009 for iHeart.



(c) Week 26-2010 for iHeart.



(d) Week 34-2009 for iSmile.



(e) Week 34-2009 for Hugged.

Figure 5: Simulated degree structures with 4 parameters.

		4 Parameter Model				6 Parameter Model			
App.	**Period**	KS_{in}	L_{in}	KS_{out}	L_{out}	KS_{in}	L_{in}	KS_{out}	L_{out}
iHeart	34-2009	0.002	0.77	0.002	2.07	0.004	1.5	0.004	0.69
	50-2009	0.01	2.07	0.01	18.08	0.002	4.23	0.002	0.62
	26-2010	0.0067	3.68	0.0067	7.75	0.0069	3.12	0.007	1.44
Hugged	34-2009	0.0001	0.27	0.0001	1.1	0.0003	0.56	0.0005	2.03
iSmile	34-2009	9.6e-005	0.38	9.6e-005	1.17	0.0002	0.63	0.0003	1.78

Table 8: Error measurements for in-degree and out-degree for the weekly UAGs.

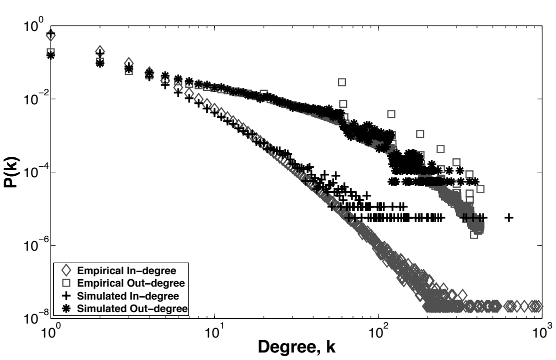


Figure 6: Simulated degree structures with 6 parameters for week 50-2009, iHeart.

Our 4-parameter model already works significantly better than existing algorithms, and the 6-parameter model works exceptionally well. We therefore do not require extensive information to model UAGs for gifting applications.

Our model potentially provides an alternative to sharing large data sets of application user activity, and since these graphs are synthetic, it also alleviates the violation of user privacy when sharing such data. Furthermore, given social applications exhibit weekly and yearly temporal patterns in user activity [16], our model is a step towards predicting user activity on other classes of social applications as well. Such a model is useful to application developers that wish to forecast an application's usage, and seek a reliable tool to help optimize growth strategies and advertising budgets. For instance, a developer might wish to buy N_0 application users from an advertising company to jumpstart (seed) their new application. Our model can provide them an approximation of user activity up to a week after application launch, which also helps them predict resource usage to minimize problems such as application downtime in the future.

5.4 Deriving Steady-State Results

Our UAG growth model allows us to run simulations given some knowledge of target graphs. However, simulating activity graphs for millions of users can be time consuming, for e.g., some of our simulations with 10 million nodes required over 5 hours to complete. We can lower our reliance on simulations through steady-state equations for degree distributions generated by our growth model. Using these equations, we can gauge statistics that convey user activity levels (such as the ratio of power to casual users) without running simulations. In this section, we probabilistically derive these distributions. We also provide a theoretical proof that our UAG growth model creates synthetic graphs with more than a single connected component.

Asymptotic Degree Distributions

Mathematical analysis allows us to obtain steady-state properties which assume the graph is infinite. However, our interest here is in producing a finite graph that captures 7 days of user activity. We find that the simulated degree distributions converge to the steady-state mathematical re-

Algorithm 1 Produces a model for weekly UAGs.

Require: N_0, M, α, β, γ_d, γ_m.
  Initialize the network G_0 as a set of N_0 isolated nodes.
  for all $x \in G_0$ **do**
    Activate x.
    $d_x \leftarrow IPL(\gamma_d, 7)$
    $m_x \leftarrow RPL(\gamma_m, M)$
    $diff_x \leftarrow \lfloor m_x d_x \rfloor - \lfloor m_x \rfloor d_x$
    $act_x \leftarrow 0$
  end for
  for $i = 0 \rightarrow 6$ **do**
    for all Active nodes x **do**
      if $diff_x \geq act_x$ **then**
        $dailyOutput \leftarrow \lfloor m_x \rfloor$
      else
        $dailyOutput \leftarrow \lfloor m_x \rfloor + 1$
      end if
      for $j = 1 \rightarrow dailyOutput$ **do**
        $coinflip \leftarrow uniform(0, 1)$
        if i = 1 and j = 1 **then**
          Add a self loop (x, x) to G_1
        else if $coinflip \leq \alpha$ **then**
          Add isolated node y to G_{i+1}
          Add the edge (x, y) to G_{i+1}
          $coinflip \leftarrow uniform(0, 1)$
          if $coinflip \geq \beta$ **then**
            Mark y for activation.
            $d_y \leftarrow IPL(\gamma_d, 7)$
            $m_y \leftarrow RPL(\gamma_m, M)$
            $diff_y \leftarrow \lfloor m_y d_y \rfloor - \lfloor m_y \rfloor d_y$
            $act_y \leftarrow 0$
          end if
        else
          Choose a random edge (y, z)
          Add the edge (x, z) to G_{i+1}.
        end if
      end for
      $d_x \leftarrow d_x - 1$
      $act_x \leftarrow act_x + 1$
      if $d_x = 0$ **then**
        Deactivate x.
      end if
    end for
    for all Marked nodes m **do**
      Activate m.
    end for
  end for
  return G_7.

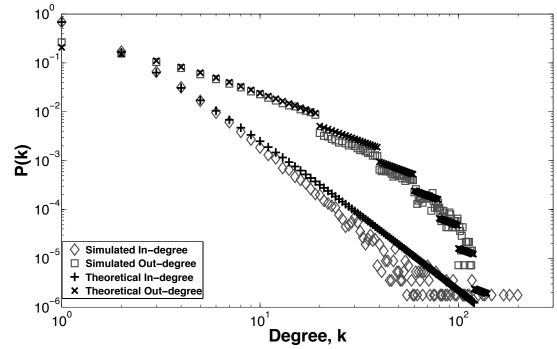


Figure 7: Simulated versus theoretical degree distributions for week 34-2009 from iHeart.

sults quickly as the number of seeding nodes (N_0) increases. Figure 7 shows example theoretical steady-state, as well as simulated, degree distributions for the week 34-2009 UAG on iHeart. Figure 7 shows that this theoretical formulation is indeed in agreement with our simulation results. The formulation is as follows:

First, consider out-degrees. Define N_t as the number of newly activated nodes at the tth timestep and N as the total number of nodes. If we assume that $x \in N_t$, where x is chosen uniformly at random, then the out-degree of x is a random variable $\lfloor m_x d_{x,t} \rfloor$ where $d_{x,t} = d_x 1_{d_x \leq 7-t} + (7-t) 1_{d_x > 7-t}$. The indicator random variable 1_A outputs 1 if the event A has occurred, and 0 otherwise. The general formula for out-degree is as follows:

$$\Pr(deg_{out}(x) = j) \quad = \quad \sum_{k=0}^{7} \frac{N_k}{N} \Pr(m_x d_{x,k} \in (j, j+1))$$

To calculate the in-degree, first define $N_{k,t}$ as the number of nodes of in-degree k after t edges have been added. If $p_{k,t}$ is the probability of observing a node of in-degree k after t edges have been added, then $p_{k,t} = \frac{t}{n} a_{k,t}$, where n is the number of nodes in the graph after t edges have been added. Dropping the t subscript in any of these variables denotes the asymptotic limit as $t \rightarrow \infty$.

$$\begin{aligned}
N_{1,t} &= N_{1,t-1} - N_{1,t-1}(1-\alpha)t - 1 + \alpha \\
N_{2,t} &= N_{2,t-1}[1 - \frac{2(1-\alpha)}{t-1} - \frac{(1-\alpha)}{n}] + \\
&\quad + N_{1,t-1}[\frac{(1-\alpha)}{t-1}] \\
&\vdots \\
N_{k,t} &= N_{k,t-1}[1 - \frac{k(1-\alpha)}{t-1} - \frac{(1-\alpha)}{n}] + \\
&\quad + N_{k-1,t-1}[\frac{(k-1)(1-\alpha)}{t-1}]
\end{aligned}$$

Using these master equations, we derive a recursive formula for p_k shown as Equation (1), which can be used to calculate p_k in time linear in k, and can also be approximated in closed form. This approximation is found by taking logs of both sides of Equation (1) and by replacing the resultant sum with an integral. This also shows that for large k, p_k decays as a PL with exponent $1 + \frac{1}{1-\alpha}$. Note that Equation (1) allows exponents strictly larger than 2.

$$\begin{aligned} p_1 &= \frac{t}{n}\frac{\alpha}{1+(1-\alpha)} = \\ &= \frac{t}{N_0+(1-\alpha)*t}\frac{\alpha}{1+(1-\alpha)} = \\ p_k &= \frac{t}{n}(p_{k-1})\frac{k-1}{\frac{1}{(1-\alpha)}+k} = \\ &= \frac{t}{N_0+(1-\alpha)*t}(p_{k-1})\frac{k-1}{\frac{1}{(1-\alpha)}+k} \end{aligned} \tag{1}$$

Connectivity

Although our model relies on a modified form of Preferential Attachment, unlike graphs with pure Preferential Attachment, our synthetic graphs are disconnected with some very small components, as in our empirical UAGs. The theorem below shows that with high probability, the synthetic graph produced is not entirely connected, and in fact for sufficiently large N_0 has a component of any fixed size.

PROPOSITION 1. *Given any fixed integer t^*, the graph G_{t^*} has multiple weakly connected components.*

PROOF. First note that $N_k \leq (M+1)N_{k-1}$, so that $N_{t^*} \leq (M+1)^{t^*} N_0$. It follows that $\frac{N_0}{N_{t^*}} \geq \frac{1}{(M+1)^{t^*}} > 0$ and thus N_0 is proportional to N_{t^*} for any choice of parameters. Let x be a node in G_t, and suppose y is a node that is sending out a heart. Since all nodes in G_t have in-degree at least 1, the probability that y sends the heart to x is at most $\frac{1}{N_t}$. Since $0 < \frac{N_0}{N_{t^*}} \leq \frac{N_t}{N_{t^*}}$ it follows that the probability that a node x of in-degree 1 receives a heart is $O(\frac{1}{N_{t^*}})$ at any given time step. The out-degree of any node is bounded by the constant Mt^*, and so the probability that x receives no hearts at all throughout the process is $(1 - O(\frac{1}{N_{t^*}}))^{Mt^* N_{t^*}} = \Omega(e^{-Mt^*})$. It follows that some non-vanishing portion of nodes in N_t receive no hearts after time $t-1$, and some non-vanishing portion of nodes in N_0 receive no hearts from other users.

Moreover, there is some non-vanishing portion of nodes in N_0 which have out-degree 1 and are connected to a node with out-degree 0. If both nodes receive no other hearts throughout the process, which will occur with probability $\Omega(e^{-2Mt^*})$, they form a component of size 2. Therefore, with high probability a portion of the nodes in N_0 will be contained in a component of size 2. Similarly it can be shown with high probability that for any $k = o(\log N_{t^*})$ there is at least on component of size k for sufficiently large n. $\square$

6. CONCLUSION AND FUTURE WORK

We have provided insight into user activity on Facebook-based gifting applications, and have presented our methodology to create the first UAG growth model for a class of social applications that use only ARs for growth. Our algorithm requires little information to produce synthetic graphs for gifting applications, and provides a feasible alternative to sharing large data sets that may violate user privacy. Together with our steady-state equations for the synthetic degree structure, our algorithm may serve as an efficient user activity prediction tool to application developers.

Our work is a first step towards modeling user activity from social applications, and as such it does suffer from some shortcomings. Specifically, our growth model does not currently differentiate between male and female users, even though we speculate there are differences in gender-wise user behavior, and it only simulates weekly UAGs without tying multiple weeks' graphs into, say, a monthly or yearly UAG. A further (albeit minor) issue is that the very low clustering of nodes in our empirical UAGs is not accurately represented in our synthetic graphs, where the clustering is in fact even lower. It is important for future work to resolve these issues to create a more flexible UAG growth model.

We wish to create an algorithm that simulates UAGs from non-gifting social applications as well. While the gifting genre of applications is the second largest on at least Facebook, gaming applications are the largest and most lucrative. This is due to the fact that social gaming itself is now a 5+ billion dollar industry, and shows no signs of declining with the advent of mobile platforms such as iPhone and Android. Modeling UAGs from *gaming* applications is, however, a highly challenging task due to use of growth mechanisms in addition to ARs (Section 3.1), and the wide variety of possible in-game activities that affect their growth.

A cursory glance at the top ten Facebook gaming applications reveals most, if not all, games incorporate a select few mechanisms to drive growth and increase user engagement. Social games differ from their counterparts in the same genre due to the extent of importance a given mechanism is afforded. Mechanics such as leader boards, periodic item give-aways, holiday-themed in-game items and non-friend interactions drive user engagement and subscribership growth, while pure growth mechanisms include gifting in-game items to OSN-resident friends. The common use of engagement and growth mechanisms in social games hints at a possible general growth model for social games.

A successful model for gaming applications must incorporate characteristics of social gaming user activity such as high clustering coefficients, presence of strong community structures, high average time spent per user, and lower organic subscriber growth compared to gifting applications [16]. For now, quantitative measurements from non-gifting applications are lacking, and this prevents research into creating a more elaborate model.

Our work is the first step towards modeling social applications that use a combination of growth mechanisms, and it begins to alleviate the lack of availability of social application data to the research community. We hope this work will encourage research into activity models for non-gifting social applications as well.

Acknowledgements

We thank Manakki, LLC for the donation of anonymized activity data. This work is supported in part by AT&T Labs–Research, Google Research Award, Defense Threat Reduction Agency, Basic Research Award No. HDTRA1-10-1-0088, and the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053.

7. REFERENCES

[1] Backstrom, L., et al. Four degrees of separation. In *ArXiv, Social and Information Networks* (2011).

[2] Cha, M., et al. Characterizing social cascades in flickr. In *ACM SIGCOMM Workshop on Online Social Networks (WoSN)* (2008).

[3] Chierchetti, F., et al. Models for the compressible web. In *Proc. of the Symposium on Foundations of Computer Science* (2009).

[4] Clauset, A., et al. Power-law distributions in empirical data. In *SIAM Review* (2009).

[5] Gjoka, M., et al. Poking facebook: Characterization of osn applications. In *ACM SIGCOMM Workshop on Online Social Networks (WoSN)* (2008).

[6] Gjoka, M., et al. Walking in facebook: A case study of unbiased sampling of osns. In *Proc. IEEE INFOCOM* (2010).

[7] Golder, S., et al. Rhythms of Social Interaction: Messaging within a Massive Online Network. In *International Conference on Communities and Technologies* (2007).

[8] Guo, F., et al. Recovering temporally rewiring networks: model-based approach. In *Proc. of International Conference on Machine Learning* (2007).

[9] Handcock, M. S., et al. Model-based clustering for social networks. In *Journal of the Royal Statistical Society, Series A* (2007).

[10] Holme, P., and Saramäki, J. Epidemic spreading in scale-free networks. In *Physical Review* (2011).

[11] Krishnamurthy, B. A measure of online social networks. In *Proc. International Conference on Communication Systems and Networks* (2009).

[12] Krishnamurthy, B., et al. On the leakage of personally identifiable information via online social networks. In *ACM SIGCOMM Workshop on Online Social Networks (WoSN)* (2009).

[13] Leskovec, J., et al. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *International Conference on Knowledge Discovery and Data Mining* (2005).

[14] Li, L., et al. A first principles approach to understanding the internet's router-level topology. In *Proc. ACM SIGCOMM* (2004).

[15] Mislove, A. and M. Marcon and K. Gummadi and P. Druschel and B. Bhattacharjee Measurement and Analysis of Online Social Networks. In *Proc. ACM IMC* (2007).

[16] Nazir, A., Raza, S., and Chuah, C.-N. Unveiling facebook: A measurement study of social network based applications. In *Proc. ACM IMC* (2008).

[17] Nazir, A., Raza, S., Gupta, D., Chuah, C.-N., and Krishnamurthy, B. Network level footprints of facebook applications. In *Proc. ACM IMC* (2009).

[18] Newman, M. E. J. The structure and function of complex networks. In *SIAM Review* (2003).

[19] Sala, A., et al. Measurement-calibrated graph models for social network experiments. In *Proc. of World Wide Web Conference* (2010).

[20] Sharan, U., and Neville, J. Temporal-relational classifiers for prediction in evolving domains. In *Proc. of International Conference on Data Mining* (2008).

[21] Tang, J., et al. Temporal distance metrics for social network analysis. In *ACM SIGCOMM Workshop on Online Social Networks (WoSN)* (2009).

[22] Torkjazi, M., Rejaie, R., and Willinger, W. Hot today, gone tomorrow: On the migration of myspace users. In *ACM SIGCOMM Workshop on Online Social Networks (WoSN)* (2009).

[23] Ugander, J., Karrer, B., et al. The anatomy of the facebook social graph. In *ArXiv, Social and Information Networks* (2011).

[24] Valafar, M., et al. Beyond friendship graphs: a study of user interactions in flickr. In *ACM SIGCOMM Workshop on Online Social Networks (WoSN)* (2009).

[25] Viswanath, B., Mislove, A., Cha, M., and Gummadi, K. P. On the evolution of user interaction in facebook. In *ACM SIGCOMM Workshop on Online Social Networks (WoSN)* (2009).

[26] App data. http://appdata.com/.

[27] Developer analytics. http://developeranalytics.com/.

[28] Farmville. https://apps.facebook.com/onthefarm/.

[29] Wilson, C., Boe, B., et al. User interactions in social networks and their implications. In *ACM EuroSys* (2009).