Title

Analyzing the Adoption and Cascading Process of OSN-based Gifting Applications: An Empirical Study

Authors
Rahman, Mohammad R
Han, Jinyoung
Chuah, Chen-Nee

Publication Date
2016

Analyzing the Adoption and Cascading Process of OSN-based Gifting Applications: An Empirical Study[1]

M. Rezaur Rahman, Department of Computer Science, University of California, Davis
Jinyoung Han, Department of Electrical and Computer Engineering, University of California, Davis
Yong Jae Lee, Department of Computer Science, University of California, Davis
Chen-Nee Chuah, Department of Electrical and Computer Engineering, University of California, Davis

To achieve growth in user base of OSN-based applications, word-of-mouth diffusion mechanisms such as user-to-user invitations are widely used. This paper characterizes the adoption and cascading process of OSN-based applications that grow via user invitations. We analyze a detailed large-scale dataset of a popular Facebook gifting application, iHeart, that contains more than 2 billion entries of user activities generated by 190 million users during a span of 64 weeks. We investigate: (1) how users invite their friends to an OSN-based application, (2) how application adoption of an individual user can be predicted, (3) what factors drive the cascading process of application adoptions, and (4) what are the good predictors of the ultimate cascade sizes. We find that sending or receiving a large number of invitations does not necessarily help to recruit new users to iHeart. We also find that the average success ratio of inviters is the most important feature in predicting an adoption of an individual user, which indicates that the effectiveness of inviters has strong predictive power with respect to application adoption. Based on the lessons learned from our analyses, we build and evaluate learning-based models to predict whether a user will adopt iHeart or not. Our proposed model that utilizes additional activity information of individual users from other similar type of gifting applications can achieve high precision (83%) in predicting adoptions in the target application (i.e., iHeart). We next identify a set of distinctive features that are good predictors of the growth of the application adoptions in terms of final population size. We finally propose a prediction model to infer whether a cascade of application adoption will continue to grow in the future based on observing the initial adoption process. Results show that our proposed model can achieve high precision (over 80%) in predicting large cascades of application adoptions. We believe our work can give an important implication in resource allocation of OSN-based product stakeholders, e.g., via targeted marketing.

1. INTRODUCTION

The word-of-mouth information diffusion [Rodrigues et al. 2011] over online social networks (OSNs) has been regarded as an important mechanism by which people discover and share a new idea, technology, content, or product. Such diffusion mechanism has gained great attentions not only from research communities but also from enterprises interested in growing their business. According to the New York Times article, traditional retailers like `Target` and `Walmart` have sought partners in OSN companies [Harris 2013].

The growing interest in such information diffusion mechanism has attracted the research community to investigate how information (e.g., a photo, link, or product) are *reshared* in OSNs like Facebook, Twitter, Flickr, or Pinterest [Cheng et al. 2014; Cha et al. 2009; Kwak et al. 2010; Rodrigues et al. 2011; Han et al. 2014]. A news, photo, link, or product may get reshared (e.g., retweet in Twitter or repin in Pinterest) multiple times over the OSNs, hence generating a *cascade* that potentially reaches a large number of users. These studies have revealed valuable insights into the macro-level propagation patterns of messages [Kwak et al. 2010], URLs [Rodrigues et al. 2011], news [Lerman and Ghosh 2010], and photos [Cha et al. 2009; Han et al. 2014]. However, most of these studies paid little attention to how *user recommendation of products* leads to their actual adoption and further propagation. This is the key to understanding and predicting the growth, popularity, and longevity of OSN-based products and/or the ultimate size of the user base.

[1] This paper is an extended version of our earlier work [Rahman et al. 2015].

In this paper, we strive to shed lights on these issues by performing a detailed investigation of the adoption process of a gifting application on Facebook. In particular, we seek answers to the following questions: How do people invite their friends to use an OSN-based application? What factors drive a new user to adopt an invitation? How often does a recruited user recruit others to the application? What are the most informative features of users in predicting application adoptions? Overall, how does the application adoption evolve over the OSN? What are the main drivers that determine a large, sustainable cascade of a new application? Can we predict the growth of the cascading process of an application?

It has been reported that Facebook gifting applications only rely on user-invitations (through sending gift items) to recruit new users [Nazir et al. 2012]. Hence, we believe that by focusing on the gifting genre, we can minimize the effect of exogenous factors, including user recruitment through other channels such as email broadcasts and paid advertisement. As a case study, we examine the launching and spreading of *iHeart*, which was one of the top gifting applications in Facebook in terms of number of (monthly) active users as of Dec. 2009. A user in *iHeart* can send a gift to other existing iHeart users, or send a gift with an invitation message (referred to as *invitation*) to his/her friends who have not installed the application yet. In the latter case, the friend who has received the invitation can either accept (i.e., install the application) or ignore it. Note that a user can receive multiple invitations from different friends until he/she accepts the application. If a user accepts the invitation and gets activated, he/she can also send invitations to other friends. Finally, these inter-user communications through invitations form a sequence of adoptions generated by a contagion process where a node "recruits" its connected nodes with some probability. We refer to this sequence of adoptions as a *cascade*.

We analyze a detailed large-scale dataset that contains more than 2 billion entries of user activities generated by 190 million users during a span of 64 weeks. Using the dataset, we first investigate (1) how users invite their friends to adopt an OSN-based application and (2) what distinctive features of users (e.g., number of sent invitations) are associated with the adoption of the OSN-based application. Note that our work focuses on analyzing how direct user-to-user invitation ('activity') leads to the actual adoption (and further propagation) of OSN-based gifting applications, whereas previous work mostly focused on how underlying friendship networks are associated with the adoption of viral online products [Katona et al. 2011; Aral et al. 2009; Aral and Walker 2012]. Based on the lessons learned from our analysis, we build and evaluate learning-based models to predict whether a user will adopt iHeart or not. We employ two additional datasets for learning: *Hugged*, which consists of 114 million entries of user activities generated by 35 million users during a span of 69 weeks, and *iSmile*, which consists of 295 million entries of user activities generated by 93 million users during a span of 128 weeks. These popular Facebook gifting applications are similar to iHeart, and we use them to investigate whether user behaviors in similar applications can be used or not to predict the user adoptions in the same genre. We next investigate (1) what factors drive the cascading process of application adoptions and (2) distinctive features associated with different sizes of cascades. Based on our findings, we develop and evaluate a prediction model to infer whether a cascade of application adoption will continue to grow in the future, which gives an important implication in resource allocation of OSN-based product stakeholders, e.g., via targeted marketing.

We highlight the main contributions and key findings of our work as follows:

(1) **Invitation and adoption behaviors – Section 4**
We found that sending or receiving a large number of invitations does not necessarily help to recruit new users, which may be due to the spamming phenomenon

where users end up ignoring the invitations. We revealed that the average success ratio of inviters is the most important feature, meaning that *capability or effectiveness of inviters* has strong predictive power with respect to application adoption. We also showed that the features related to invitations-receiving patterns are also good indicators in predicting application adoptions.

(2) **Predicting an application adoption of an individual user – Section 5**
Based on the lessons learned from our analysis, we build and evaluate learning-based models to predict whether a user will adopt iHeart or not. To investigate whether the additional activity information of individual users in other similar applications (Hugged and iSmile in our case) can help to enhance the predictability of their adoptions in our target application (iHeart), we identified the active user population that is 'common' across the three applications and learned additional information of each user observed in Hugged and iSmile: (i) adoption information whether or not he/she adopts Hugged and iSmile and (ii) his/her activity information in Hugged and iSmile. We showed that the precision of the proposed model reach to 83%, which demonstrate that using past behavioral information of each corresponding user in similar applications can significantly enhance the predictability of application adoption.

(3) **Growth patterns of application adoptions (or cascades) – Section 6**
We analyzed the growth patterns of the application adoption process in iHeart from a graph-theoretical perspective. We found a set of distinctive features (e.g., structural or evolutionary properties of cascades, and roles of their seed nodes) that are good predictors for the final population size of the cascade. To attain large cascades (e.g., beyond 100 users), effective user recruitment has to continue beyond the initial growth phase. In general, we observed that early adopters (seeds and their recruited users) play a diminishing role as cascade sizes increase.

(4) **Predicting large cascades of application adoptions – Section 7**
We explored the implications of our findings for predicting the attainable growth of different application adoption cascades. We found that the evolutionary properties of cascades (e.g., initial growth rate) are good predictors of the ultimate cascade sizes. By observing these features during the initial adoption process, our classifier can achieve over 80% precision for predicting the large cascades of application adoption. Our learning-based prediction model performs well when applied to another gifting application, *Hugged*, which suggests that our prediction model is generally applicable to gifting genre of OSN-based applications where the new users are recruited primarily through user invitations.

The rest of the paper is organized as follows. After reviewing related work in Section 2, we describe our datasets in Section 3. We then characterize the application adoption process of iHeart, and describe the potential factors that drive the application adoptions in Section 4. Section 5 evaluates the proposed learning-based models to predict whether a user will adopt iHeart or not. We next present our measurement-based characterization of application cascade process in Section 6. Based on insights learned from our analyses, we build and evaluate a learning-based prediction model to forecast the growth of the cascading process in Section 7. Finally, we conclude the paper in Section 8.

2. RELATED WORK

Information adoption and propagation in OSNs: As OSNs have become the norm to spread information such as news, images, or video URLs, there have been many studies that investigate the patterns of information adoption in OSNs, revealing valuable insights into characteristics of information spreading [Rodrigues et al. 2011; Han et al. 2015; Goel et al. 2012; Cha et al. 2009; González-Bailón et al. 2011; Bakshy et al. 2012; Cha et al. 2010; Wang et al. 2011; Han et al. 2014; Kitsak et al. 2010]. Rodrigues *et al.* analyzed the patterns of word-of-mouth URL exchanges in Twitter, and found that users with geographically similar locations tend to share the same URL [Rodrigues et al. 2011]. **González-Bailón *et al.* [González-Bailón et al. 2011] studied protest recruitment patterns in Twitter, which revealed evidence for social influence and complex contagion.** Han *et al.* investigated the gender differences in sharing images (or pins) on different topics (e.g., travel, education, food), and showed that male and female users show different behaviors on different topics in terms of dedication, responsiveness, and sentiment [Han et al. 2015]. Goel *et al.* [Goel et al. 2012] investigated the cascades of URLs shared in different micro-blogging services such as Yahoo! or Twitter. They showed that multi-step cascading is not frequently observed; the vast majority of the diffusions occur within one hop from a seed node. Similar characteristics are also observed in photo-sharing OSNs like Flickr. Cha *et al.* [Cha et al. 2009] reported that even popular photos do not spread widely in Flickr. Our work focuses on how *OSN-based applications* are recommended, adopted, and propagated in the OSNs, which is the key to understand the growth, popularity, and longevity of OSN-based applications or products.

Some of the studies have focused on what drives information diffusion in OSNs. There have been increasing attention to effectively identify special individuals, often referred to as the 'influentials' [Berry and Keller 2003] who can recruit a large number of users, leading to a large-scale cascade [Watts and Dodds. 2007]. Bakshy *et al.* showed that weak ties play important roles in spreading new information on Facebook [Bakshy et al. 2012]. Cha *et al.* showed that the most influential users who generate a large number of retweets are not necessarily the most followed (or popular) ones in Twitter [Cha et al. 2010]. Han *et al.* also showed that the potential influential users who have a large number of followers may not generate a large number of repins in Pinterest [Han et al. 2014]. **Similarly, Kitsak *et al.* [Kitsak et al. 2010] found that the best spreaders are not necessarily the most highly connected or the most central people in various social networks (e.g,. LiveJournal, Email, or IMDB).** Wang *et al.* showed that social and organizational context impacts on information diffusion in OSNs [Wang et al. 2011]. These studies reveal valuable insights into (i) how users adopt and propagate information and (ii) what are the key drivers to spread information such as URLs, news, or images on various OSNs. In contrast, this paper focuses on (i) how invitation-based online social applications (or products) are recommended and adopted and (ii) what factors drive the adoption of new OSN-based applications.

Online product adoption and diffusion in OSNs: There have been efforts in investigating how online products (e.g., an OSN-based application or a new OSN) are adopted/propagated [Katona et al. 2011; Aral et al. 2009; Aral and Walker 2012; Jeong and Moon 2014]. Katona *et al.* analyzed the diffusion of an online social network, and modeled the adoption decision of individuals by observing (i) the local network structure formed by already adopted neighbors, (ii) the average characteristics of adopted neighbors, and (iii) the characteristics of the potential adopters [Katona et al. 2011]. They revealed that an individual connected to many (already) adopters (in his/her friend relationships) shows a higher adoption probability. Aral *et al.* developed a dynamic matched sample estimation framework to distinguish peer-influence and homophily [McPherson et al. 2001] effects in dynamic networks, and they applied their proposed framework to a five month dataset of the day-by-day adoption of a mobile service application [Aral et al. 2009]. They found that previous methods overestimate peer-influence in product adoption decisions. Aral *et al.* also analyzed and identified influential individual and susceptible members in adopting the offered product on a representative sample of 1.3 million Facebook users [Aral and Walker 2012]. **Jeong and Moon analyzed the spread of mobile games via a popular mobile instant message service in Korea, called KakaoTalk [Jeong and Moon 2014].**

While these studies mostly focused on the underlying friendship networks to analyze the adoption patterns of viral online products, we instead focus on the gifting genre of Facebook applications, as the gifting genre only relies on invitations to recruit new users from Facebook [Nazir et al. 2012], to investigate how user recommendation of OSN-based products through invitations leads to their actual adoption. In this way, we believe that the effect of exogenous factors (e.g., user recruitment through other channels such as email broadcasts or paid advertisement) for application adoptions can be minimized.

Popularity or cascade size prediction: There have been great interests in predicting the future popularity of a given content, information, or product to increase sales. For example, content or web service providers like YouTube may want to estimate the view counts (or download counts) of their content in advance, which are directly tied to their *ad* revenue [Szabo and Huberman 2010]. This has led to the investigations of the popularities of Digg stories [Szabo and Huberman 2010], information on Twitter [Yang and Leskovec 2010], and YouTube view counts [Szabo and Huberman 2010]. Also, several studies have focused on how to predict the growth of information cascading in Twitter [Kupavskii et al. 2012] and Facebook [Cheng et al. 2014]. Cheng *et al.* showed that temporal and structural features are key predictors of the size of photo reshare cascade in Facebook [Cheng et al. 2014]. In this paper, we predict the success/failure of a seed node (or early adopters) to create a large-scale cascade of application adoption using a learning-based model and insights gained from our measurement studies. Such model has great utility in resource allocation of product stakeholders, e.g., via targeted marketing.

User interactions via Facebook applications: Facebook has become one of the most popular online social networks (OSNs) over the last decade. With its user base of over one billion, many software companies have launched their applications on Facebook, which has reached a $6+ billion industry (as of 2012) [Nazir et al. 2012]. For example, *Zynga*, a popular social game company, launched games on Facebook that had over 265 million monthly active users [app 2013], and 80% of their revenue came from Facebook users (as of Jan. 2013) [bbc 2012]. This has motivated numerous studies on the various aspects of Facebook applications [Gjoka et al. 2008; Kirman et al. 2009; Wei et al. 2010; Li et al. 2013; Liu et al. 2013; Rahman et al. 2014]. Gjoka *et al.* found that the popularities of Facebook applications exhibit a highly skewed distribution [Gjoka et al. 2008]. Kirman *et al.* studied the social structures of Facebook games [Kirman et al. 2009]. Wei *et al.* examined the roles of users to propagate Facebook gaming applications [Wei et al. 2010]. Li *et al.* studied an efficient recommendation system that considers the popularity and reputation of an application as well as the preferences and social relationships of a user [Li et al. 2013]. Liu *et al.* developed a model of the evolution trend of Facebook gifting applications [Liu et al. 2013].

Our preliminary work investigated the role of the seeds in the adoption process of the Facebook gifting application, and showed that large-scale adoptions are not correlated to well-known properties of seeds (such as outdegree or active lifetime) [Rahman et al. 2014]. This paper goes one step

Table I. Summary of the three datasets: Hugged, iSmile, and iHeart.

App.	Launched	Data duration	Total number of users	Number of activity entries
Hugged	Feb. 2008	69 weeks	35 M	114 M
iSmile	Aug. 2008	128 weeks	93 M	295 M
iHeart	June 2009	64 weeks	190 M	2 B

further; we investigate (i) how user recommendation of Facebook applications (through invitations) leads to their actual adoption and (ii) what factors drive the adoption and propagation of Facebook applications. In addition, we apply the insights learned from our measurement study to develop a learning-based model to predict whether a cascade of application adoption will continue to grow in the future.

3. DATASETS

We conduct a measurement study on how OSN-based gifting applications are recommended, adopted, and propagated over Facebook platform. We analyze detailed datasets of three popular Facebook gifting applications, *Hugged*, *iSmile*, and *iHeart*, which were operated by Manakki, LLC at the time of data collections. **Hugged, iSmile, and iHeart are home-grown Facebook applications developed by a former member of our group in collaboration with Manakki. All the time-stamped user activity records in the applications were logged at the application servers. Each user activity record includes the sender's and recipient's anonymized Facebook user IDs as well as the time stamp when the sender sends an invitation to the recipient. Note that we make our anonymized datasets online at: http://rubinet.ece.ucdavis.edu/data.html.** In this paper, we do not consider all the interactions among users (including sending and receiving gifts), but we focus on analyzing 'invitations' sent to the new users until they first become activated.

Table I summarizes the three collected datasets, Hugged, iSmile, and iHeart:

- **Hugged: Users in Hugged (launched in February 2008) share hugging-related graphics as gifts with their Facebook friends. 114 M users either send or receive at least one gift during our measurement period. Among them, only 35 M users are active senders (i.e., sends at least one gift), which implies that they have installed (or adopted) Hugged.** The Hugged dataset covers the 69-weeks lifespan of the application from March 2008 and contains 114 M entries of user activities generated by 35 M users.
- **iSmile: Users in iSmile (launched in August 2008) share smiling characters as gifts with their Facebook friends. 295 M users either send or receive at least one gift during our measurement period. Among them, only 93 M users are active senders (i.e., sends at least one gift), which implies that they have installed (or adopted) iSmile.** The iSmile dataset covers 128-weeks lifespan of the application from June 2009 and contains 295 M entries of user activities generated by 93 M users.
- **iHeart: Users in iHeart (launched in June 2009) share heart-shaped characters as gifts with their Facebook friends. 190 M users either send or receive at least one gift during our measurement period. Among them, only 76 M users are active senders (i.e., sends at least one gift), which implies that they have installed (or adopted) iHeart.** The iHeart dataset covers the 64-weeks lifespan of the application from June 2009 and contains 2 B entries of user activities generated by 190 M users.

iHeart was one of the top three Facebook applications in terms of number of monthly active users as of Dec. 2009. We first use the iHeart dataset to characterize (i) how an OSN-based application is recommended and adopted by new users, and (ii) what factors drive the adoption process (in Section 4). We then build and evaluate prediction models to predict user adoptions of iHeart (in Section 5). To investigate whether user behaviors in other similar type applications (Hugged and iSmile) can enhance the predictability of application adoption of iHeart users, we employ the Hugged and iSmile datasets. Note that, all three applications are shared by around 16.5 million users.

We next characterize the cascade process of iHeart adoptions (in Section 6). To analyze the adoption process of iHeart, we identified the *seed* nodes, i.e., users who already installed and used the applications without receiving any prior invitations/gifts from other users. We manage to uncover 2,384,686 cascades (i.e., cascade sizes are bigger than 1), each rooted at a unique seed. Together, these cascades contain 186,254,526 recruited iHeart users. Note that there are another 1,350,095 users who installed iHeart without prior invitations but have not recruited any users in our dataset, and hence are excluded from our study. We then build and evaluate learning-based prediction models to forecast the growth of the cascading process of iHeart adoptions (in Section 7). In addition, we further use the Hugged dataset for the purpose of verifying the universality of our proposed prediction model. We analyze 461,510 cascades generated by unique seeds, which contain 33,967,052 recruited Hugged users. Note that there are another 97,131 seed nodes that fail to recruit any users and are excluded from our study.

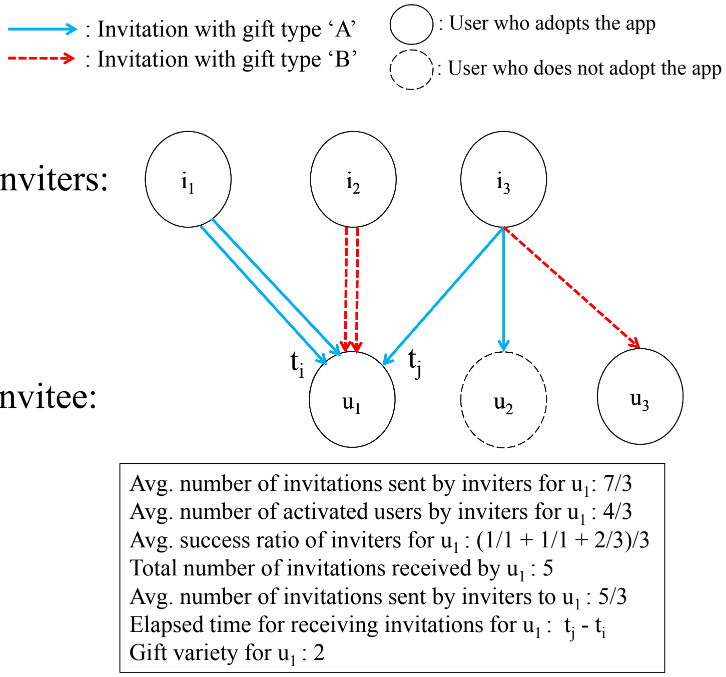
4. CASE STUDY: INVITATION-BASED ADOPTION AND PROPAGATION OF IHEART

In this section, we characterize the adoption process of iHeart, and describe the potential factors that drive application adoption of iHeart users. An iHeart user sends invitations to recruit his/her friends and some of them may accept the invitations. This results in an inviter-invitee relationship. For example, if user A sends an invitation to a friend B (who has not yet adopted iHeart), A becomes the inviter of B and B becomes the invitee of A. An invitee (user B) can receive multiple invitations from multiple inviters (including user A) until he/she accepts them and adopts the application. Figure 1 illustrates an inviters-invitees relationship generated by multiple inviters and invitees. The values of the features (that will be described later) in the box are calculated for this example. In the following subsections, we explore (i) invitation patterns of inviters and (ii) adoption behaviors of invitees, both of which are key to the successful growth of iHeart user base.

4.1. Invitation patterns of inviters

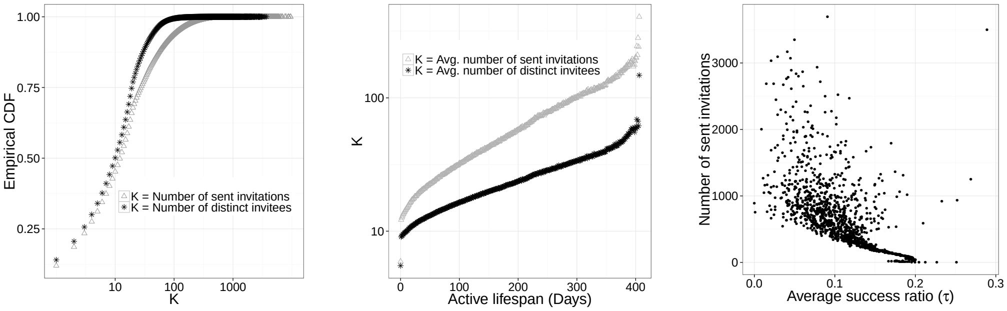
We first show the distributions of the number of invitations sent by an inviter and the number of unique invitees targeted by the inviter in Figure 2(a), respectively. We find that more than 50% of the inviters send invitations to 10 or more distinct invitees. Moreover, 50% of the inviters send more than 13 total invitations. Note that the top 1% of inviters (in terms of number of sent invitation) send more than 84 invitations.

To investigate how long an inviter is active (i.e., using iHeart to send invitations to recruit new users), we calculate the *active lifespan* of each inviter as the time difference between the first and the last invitation sent. During the active lifespan, an inviter may send multiple invitations. We investigate the correlation between the active lifespan of the inviters and the number of invitations that are sent to their friends in Figure 2(b). As shown in Figure 2(b), the inviters with longer active lifespan send more invitations and invite more unique invitees. For instance, inviters with longer than 300 days of active lifespan send more than 100 invitations on average. Notice


: Invitation with gift type 'A'
: Invitation with gift type 'B'
: User who adopts the app
: User who does not adopt the app
Inviters:
i_1
i_2
i_3
t_i
t_j
Invitee:
u_1
u_2
u_3

Avg. number of invitations sent by inviters for u_1: 7/3
Avg. number of activated users by inviters for u_1 : 4/3
Avg. success ratio of inviters for u_1 : (1/1 + 1/1 + 2/3)/3
Total number of invitations received by u_1 : 5
Avg. number of invitations sent by inviters to u_1 : 5/3
Elapsed time for receiving invitations for u_1 : t_j - t_i
Gift variety for u_1 : 2

Fig. 1. An example of inviters-invitees relationships. The box shows the values of the features of inviters and invitees (introduced in this section) for this example.



(a) Number of sent invitations

(b) Active lifespan of an inviter and her activities

(c) Average success ratio vs. number of invitations

Fig. 2. Characteristics of inviters in iHeart.

that the inviters with at least a day of active lifespan send more than 12 invitations to more than 9 distinct invitees on average.

We next explore the inviter's performance in terms of the success ratio τ of user i, which is defined as:

$$\tau_i = \frac{A(i)}{N(i)} \quad (1)$$

where $A(i)$ is the number of activated users who received an invitation from i, and $N(i)$ is the number of total users who received an invitation from i. We observe that 3% of the inviters exhibit τ of 0, which means they may have zero impact on the growth of iHeart. However, many of these inviters with τ = 0 (who have not succeeded in recruiting new users) send a large number of invitations (the average number of invitations sent by them is 5.37, with a maximum of 892 invitations). This indicates that sending a large number of invitations does not necessarily mean that the inviter will succeed in recruiting large number of new users. In fact, the inviters sending many invitations tend to have a low average success ratio, as shown in Figure 2(c). On the other hand, the inviters who send a small number of invitations (perhaps to more targeted recipients) show large success ratios. Note that the Pearson correlation coefficient between the number of invitations of inviters and their average success ratio is -0.612 **($p - value < 0.001$ with 95% confidence interval)**.

We now introduce distinctive features of inviters that are potentially associated with iHeart adoptions as follows:

— **Average number of invitations sent by inviters,** $IV(V(u))$ indicates the average number of invitations sent by the inviters of invitee u, where $V(u)$ is the set of distinct inviters that have sent invitations to the invitee u. Say, $p(v)$ as the number of sent invitations to any users including but not limited to user u by the inviter $v \in V(u)$, then the average number of sent invitations by inviters in $V(u)$ is,

$$IV(V(u)) = \frac{\sum_{v \in V(u)} p(v)}{|V(u)|} \tag{2}$$

For example, in Figure 1, $V(u_1) = \{i_1, i_2, i_3\}$, $p(i_1) = 2$, $p(i_2) = 2$, $p(i_3) = 3$, and hence $IV(V(u_1)) = \frac{2+2+3}{3} = \frac{7}{3}$. The average number of invitations sent by inviters for u_1 is 7/3.

— **Average number of activated users by inviters,** $AU(V(u))$ estimates the capabilities of inviters to recruit new users to the application. Let's say $a(v)$ as the number of activated users who adopt the application after receiving one or more invitations from the inviter v. Then the average number of activated users by inviters of u is,

$$AU(V(u)) = \frac{\sum_{v \in V(u)} a(v)}{|V(u)|} \tag{3}$$

For example, in Figure 1, $V(u_1) = \{i_1, i_2, i_3\}$, $a(i_1) = 1$, $a(i_2) = 1$, $a(i_3) = 2$, and hence, $AU(V(u_1)) = \frac{1+1+2}{3} = \frac{4}{3}$. The average number of activated users by inviters for u_1 is 4/3.

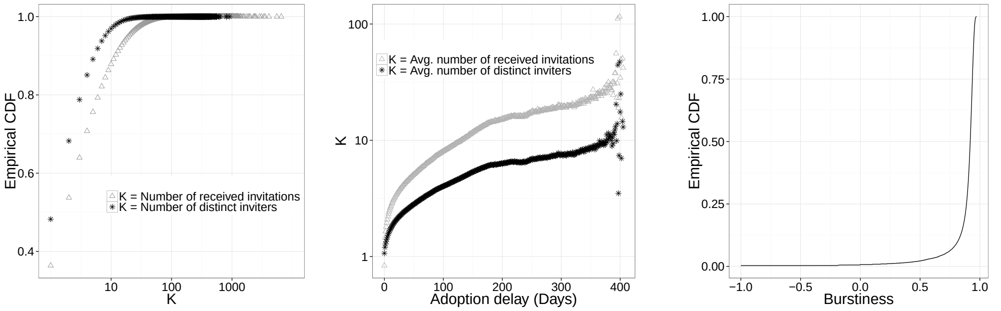
— **Average success ratios of inviters,** $SR(V(u))$ indicates the average success ratios that inviters have. Here, the success ratio $\tau(v)$ of an inviter v (introduced in Eq. 1) represents the effectiveness of the inviter to recruit new users to the application. The average success ratios of inviters is,

$$SR(V(u)) = \frac{\sum_{v \in V(u)} \tau(v)}{|V(u)|} \tag{4}$$

For example, in Figure 1, $V(u_1) = \{i_1, i_2, i_3\}$, $\tau(i_1) = 1/1$, $\tau(i_2) = 1/1$, $\tau(i_3) = 2/3$, and hence, $SR(V(u_1)) = (1/1+1/1+2/3)/3 = \frac{8}{9}$. The average success ratio of inviters for u_1 is 8/9.

4.2. Adoption behaviors of invitees

We next investigate the adoption behaviors of invitees. Figure 3(a) shows the distributions of the number of invitations received by an invitee and the number of unique inviters from which an invitee receives invitations, respectively. We find that 86% of the invitees receive less than 10 invitations, but 1% of invitees receive more than 46



(a) Number of received invitations (b) Adoption delay vs. receptions (c) Invitees' reception burstiness

Fig. 3. Adoption behaviors of invitees in iHeart.

invitations. We also investigate the time gap between the first time a user gets an invitation and the time a user gets activated, which we refer to as *adoption delay*. For example, if a user gets an invitation at $t1$ for the first time and she gets activated at $t2$, the adoption delay is $t2-t1$. We find that the adoption delays of 29.7% of the (activated) users are less than a day; this includes the users who promptly accept the invitations after receiving them. We also observe that most (85%) of the users have less than 100 days adoption delays, but around 2.6% of the users have adoption delays longer than 200 days. Figure 3(b) shows the correlation between the adoption delay and the average number of invitations received by an invitee. Invitees with longer adoption delays tend to receive more invitations and are targeted by more unique inviters.

We also investigate how regularly an invitee receives multiple invitations, i.e., the arrival patterns of the invitations. To this end, we use the *burstiness parameter B* [Sorribes et al. 2011; Goh and Barabási 2008], which can be calculated as:

$$B = \frac{\sigma_X - \mu_X}{\sigma_X + \mu_X} \tag{5}$$

where X is a set of number of invitations in each time slot, and σ_X and μ_X are the standard deviation and the mean of X, respectively. That is, $x_t \in X$ where $X = \{x_1, x_2, x_3, \ldots, x_t\}$ indicates that an inviter receives x_t invitations during the given time slot t. We set the time slot as 1 hour. Note that the burstiness parameter B has values in the range between -1 and 1, where $B = -1$ means periodic activities, $B = 0$ means random activities (Poissonian), and $B = 1$ means completely bursty dynamics [Sorribes et al. 2011]. We plot the CDF of the burstiness of the invitations received by the invitees in Figure 3(c). As shown in Figure 3(c), we find that burstiness values of 90% of the invitees are mostly close to 1, which means the most of invitees receive invitations in a significantly skewed fashion, e.g., in a burst or batch, without any regularity.

We next explore how many invitations (or inviters) motivate the invitees to adopt iHeart. To this end, we calculate the probability of adoption after receiving n invitations as:

$$p(n) = \frac{N_a(n)}{N_a(n) + N_{na}(n)} \tag{6}$$

where $N_a(n)$ is the number of invitees who received n invitations and got activated, and $N_{na}(n)$ is the number of the invitees who received n invitations but never got activated. Similarly, we can calculate the adoption probability $p(m)$ for m inviters by replacing

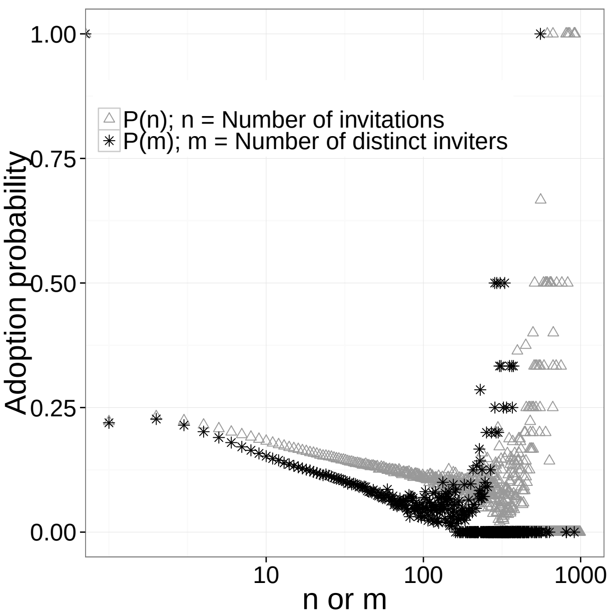


Fig. 4. Adoption probability as a function of number of invitations, n, and the number of unique inviters, m, respectively.

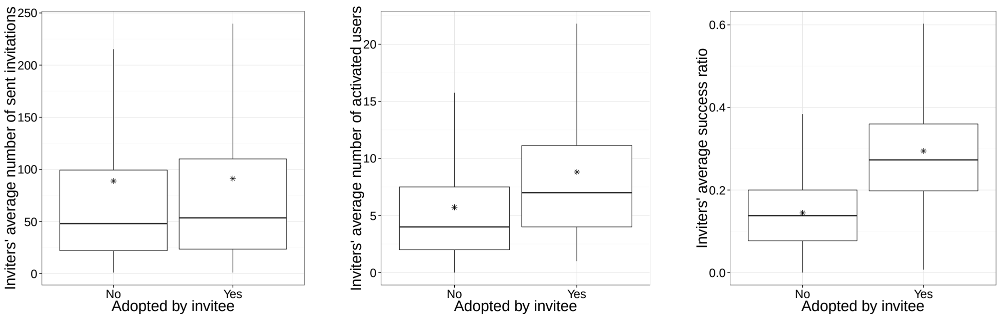
n in invitations to m inviters. We plot $p(n)$ and $p(m)$ against n and m in Figure 4, respectively, where n stands for the number of invitations, and m is the number of distinct inviters. As shown in Figure 4, $p(n)$ and $p(m)$ decrease as n and m increase. This indicates that a large number of invitations may not necessarily help in recruiting more new users to adopt iHeart; this may be due to the spamming phenomenon and an important lesson for application recruiters. When n and m go beyond 100 invitations and inviters, $p(n)$ and $p(m)$ exhibit diverse patterns, but we observe that most of the invitees who do adopt iHeart receive only around 4 invitations from 2 distinct inviters on average.

We now illustrate distinctive features of an invitee that are possibly associated with iHeart adoptions as follows:

- **Total number of invitations received by invitee u,** $i(u)$ is the total number of invitations user u receives. For example, u_1 in Figure 1 receives 5 invitations, thus $i(u_1) = 5$.
- **Average number of invitations sent by inviters to invitee u,** $i'(u)$ can be calculated as,

$$i'(u) = \frac{i(u)}{|V(u)|} \tag{7}$$

  where $|V(u)|$ represents the number of unique inviters of u. For example, in Figure 1, $i(u_1) = 5$, $|V(u_1)| = 3$, and hence, $i'(u) = \frac{5}{3}$. The average number of invitations sent by inviters to u_1 is 5/3.
- **Elapsed time for receiving invitations,** $e(u)$ is the time gap between the first and last invitations invitee u receives. For example, in Figure 1 if u_1 receives the first invitation at t_i and last invitation at t_j, then the elapsed time for receiving invitations for u_1 is $t_j - t_i$.



(a) Average number of invitations sent by inviters

(b) Average number of activated users by inviters

(c) Average success ratios of inviters

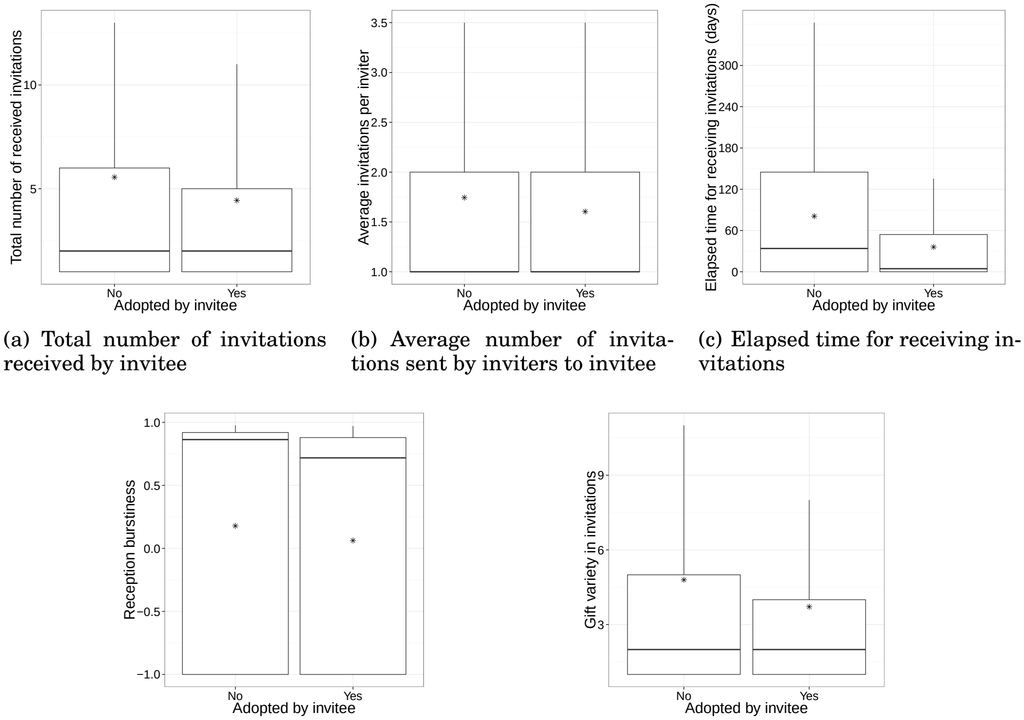
Fig. 5. Box plots showing the impact of various *inviters* features on application adoption. The $*$ indicates the mean.

- **Reception burstiness,** $b(u)$ represents how regularly an invitee u receives multiple invitations, which is defined in Eq. 5. This feature indicates the arrival patterns of invitations.
- **Gift variety,** $g(u)$ is the number of distinct gift types received by invitee u. For example, u_1 in Figure 1 receives two types of gifts (among five invitations), hence $g(u_1)$ = 2.

4.3. Adoption Patterns on Different Features

We investigate whether the above features of inviter(s) and the invitee are connected to the application adoption of the invitee in Figures 5 and 6, respectively. We first examine the three inviters features: average number of invitations sent by inviters (Figure 5(a)), average number of activated users by inviters (Figure 5(b)), and average success ratios of inviters (Figure 5(c)). We find that the invitees who adopted and did not adopt the application show similar distributions in terms of average number of invitations sent by inviters in Figure 5(a), which indicates that the activeness of inviters (i.e., number of sent invitations) is not directly associated with the application adoption. However, when we look at the *capability or effectiveness* of inviters in terms of their average number of activated users and the average success ratios in Figures 5(b) and 5(c), respectively, we find that these features are significantly associated with the application adoption. The invitees who adopted the application received the invitations from inviters who have many number of activated users or high success ratio. For example, in Figure 5(b), the median value of the average number of activated users by inviters for the users who do not adopt the application shows 4, while that for the users who adopt the application is 7. Similarly, the median value of the average success ratio of inviters for the users who do not adopt is only 0.14, whereas that for the users who adopt the application is around 0.27. This implies that capability or effectiveness of inviters has strong predictive power with respect to application adoption.

We also examine whether the invitee features are associated with application adoption or not. We find that total number of invitations received by invitee and average number of invitations sent by inviters to invitee are not connected to the application adoption in Figures 6(a) and 6(b). However, we observe that invitees who adopted and did not adopt the application show different patterns in terms of elapsed time for receiving invitations and reception burstiness in Figures 6(c) and 6(d), meaning that features related to AR-receiving patterns are good indicators to predict the application adoption. The number of invitations received by users who adopt and do not adopt the



(a) Total number of invitations received by invitee

(b) Average number of invitations sent by inviters to invitee

(c) Elapsed time for receiving invitations

(d) Reception burstiness

(e) Gift variety

Fig. 6. Box plots showing the impact of various *invitee* features on application adoption. The $*$ indicates the mean.

application are similar to each other as shown in Figure 6(a), whereas the elapsed time for users who adopt and do not adopt the application show different patterns in Figure 6(c). Note that median, average, and 75th percentile of elapsed times for receiving invitations for invitees who adopted the applications are 4.5, 36, and 54.25 days, respectively, implying that most of invitees adopt the application in less than 2 months after they receive their first AR. When we look at the burstiness in Figure 6(d), the median value of burstiness for the users who do not adopt is 0.86, while that for the users who adopt is 0.72, which indicates that users who do not adopt iHeart receive more bursty invitations than those who adopt. We finally investigate whether or not the number of gift types the invitee has (i.e., gift variety) is associated with the application adoption. As shown in Figure 6(e), the gift types of the invitees who adopted the application is not very different from those of the invitees who did not adopt the application. This implies that gift variety may not associate much to user adoption of iHeart.

In summary, capability or effectiveness of inviters along with invitee's features related to invitation-receiving patterns are good user adoption indicators.

5. PREDICTION ON APPLICATION ADOPTION

Our measurement-based characterization of the adoption patterns of OSN-based application suggest that there exists a set of distinctive features that can identify whether a user will adopt the application or not. In this section, we seek answers to the following questions: (1) How can we predict application adoption of a user? Which features (e.g. inviters features and/or invitee features) are effective to predict application adoption? (2) How does the proposed model perform in different application evolution phases (e.g., growth, peak, and decline) in predicting user adoptions? (3) Is learning from other users of similar gifting applications effective for predicting application adoption? and (4) Does activity information of individual users from other similar type of applications enhance the performance of predicting application adoption?

5.1. Prediction as a Learning Problem

5.1.1. Problem Definition. Our goal is to predict whether a user will adopt iHeart or not. We cast this as a learning problem, where we observe the characteristics and adoption patterns of the users (i.e., invitees) and their inviters and then predict whether a user will adopt iHeart or not. **Note that the numbers of users who adopt and non-adopt the iHeart are 39,403,481 and 146,851,097, respectively.**

5.1.2. The Classifier and Performance Metrics. We build a logistic regression model based on the observed features and adoption patterns. We used a variety of classifiers including linear regression, naive Bayes, Support Vector Machines (SVM), and decision trees. However, we report the performance of the logistic regression model for sake of brevity since it performs similar or better than other classifiers in most cases. Our prediction model tags each user as either adopted or not-adopted. However, the model may fail to identify user adoptions (i.e., false negatives (FN)). Conversely, it could declare not-adopted users as adopted (i.e., false positives (FP)). If functioning well, the model may correctly identify actually adopted users (i.e., true positives (TP)) and actually not-adopted users (i.e., true negatives, (TN)). Note that the performance measures that depend on the values of TP, FP, TN and FN, require a binary decision from the model. However, many of the classification techniques that are used to build models, do not yield binary decisions, rather they give the probability of an entity being adopted. We use a *cutoff* probability of 0.5 to discretize the continuous probability into a binary decision, where any entity with a adoptness probability > *cutoff* would be considered by the model as adopted, otherwise the model would consider the entity as not-adopted.

We define several commonly used performance measures that we use to evaluate the effectiveness of our model:

- **Accuracy (ACC):** is the proportion of correct predictions, which consists of both predicting adopted users as adopted and not-adopted users as not-adopted; i.e., $ACC = \frac{TP+TN}{TP+FP+FN+TN}$. Accuracy is an indicator of model capability to identify both adopted and not-adopted instances.
- **Precision (Prec):** The fraction of instances that are actually adopted among all the instances classified as adopted; i.e., $Prec = \frac{TP}{TP+FP}$. If precision is low, most of the classification is useless and not desirable in practice.
- **True positive rate (TPR) or recall:** The fraction of instances correctly classified as adopted, among all the instances that actually adopted; that is $TPR = \frac{TP}{TP+FN}$. A good model should achieve both high recall (finding most of the adopted users) and high precision (avoid too many false positives).
- **False positive rate (FPR):** The fraction of instances that are wrongly classified as adopted, among all the not-adopted instances; that is $FPR = \frac{FP}{FP+TN}$. A low FPR is desirable for a good predictor.
- **Area under ROC curve (AUC):** The Receiver Operating Characteristics curve (ROC) [roc 2006] plots the variation of false positive rate vs. true positive rate for all the instances in the test data. ROC is a non-parametric method of evaluating models, which is unaffected by class imbalance (proportion of true positives in the dataset) and also is (unlike the four measures above) independent of the cutoff. It

Table II. Feature importances based on Cramer's V association for all three applications. Note that the importances are separately computed for each application.

	$iHeart$	$Hugged$	$iSmile$
Average success ratio of inviters ($SR(V(u))$)	0.249	0.202	0.175
Average number of activated users by inviters	0.132	0.110	0.103
Elapsed time for receiving invitations ($e(u)$)	0.115	0.051	0.051
Reception burstiness ($b(u)$)	0.113	0.052	0.050
Gift variety ($g(u)$)	0.033	0.021	0.015
Total number of invitations received by invitee ($i(u)$)	0.028	0.017	0.015
Average number of invitations sent by inviters to invitee ($i'(u)$)	0.026	0.013	0.010
Average number of invitations sent by inviters ($IV(V(u))$)	0.025	0.034	0.019

represents the precision/recall pairing for all possible cutoff values between 0 and 1. In particular, ROC is a 2-D curve that plots the TPR on the y-axis and FPR on the x-axis and all such points pass through (0,0) and (1,1). The ideal ROC curve approaches the top left corner for 1 true positive rate and 0 false positive rate. A random model will be close to the diagonal y = x. The area under the ROC curve (AUC) is a measure of model performance. A perfect model will have an AUC of 1.0. A random prediction yields an AUC of 0.5. AUC is independent of threshold setting, which makes it a useful measure to compare prediction performance on different test sets.

5.2. Feature Importance

We have analyzed the eight features that can be associated with application adoption as described in Section 4. Here, we seek to identify the specific features that contribute most towards predicting user adoption. For this purpose, we apply *Chi-squared* (χ^2) statistic evaluation [Liu and Setiono 1995] to all of the mentioned features and assign a score to each one, which symbolizes the relationship between the feature and the application adoption. χ^2 statistic is used to evaluate the 'distance' between the distribution of adopted and not-adopted users for a given feature. The bigger the value, the more effective is the feature in identifying application adoption. We find very high χ^2 values for each of the candidate features, and thus rank them according to their feature importance calculated by Cramer's V [Liebetrau 1983] between each feature and users' response to adoption. Cramer's V association (defined in Eq. 8 for dichotomous adoption and training set size n) determines the strengths of association of the features after their significances are determined by the χ^2 statistic. Note that we compute the feature importance with respect to user adoptions in each application separately. Keeping that in mind, for all three applications, we report feature importance based on Cramer's V association in Table II. **Note that all the $p - values$ are lower than 0.001 with 95% confidence interval.**

$$V = \sqrt{\frac{\chi^2}{n}} \tag{8}$$

In all three applications, the most important feature is the average success ratio of inviters, which means the average success ratio of inviters is the best indicator to determine whether the invitee will adopt the application or not. The second most important feature is the average number of activated users by inviters, which also indicates that *capability or effectiveness* of inviters has strong predictive power with respect to application adoption. When we look at the next two important features, they are the invitee-related ones: the elapsed time for receiving invitations and the reception burstiness. This implies that invitation-receiving patterns are also important indicators for predicting application adoption. Notice that the top four important fea-

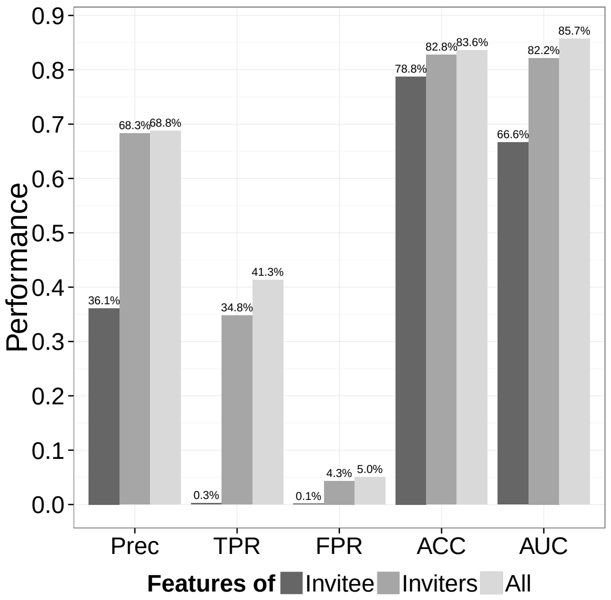


Fig. 7. Performance of models based on learning from invitee and inviters features. We test learning models (based on invitee features and inviters features) on iHeart users. Note that 'All' collectively considers both invitee and inviters features.

tures are common across the three applications, which implies that these features are useful to predict application adoption regardless of the specific application.

5.3. Performance

We now illustrate the performance of our proposed model based on the observed features.

5.3.1. Learning from Inviters and Invitee Features. We first investigate the performance of models based on different type of features. As discussed in Section 4, we consider two models: (i) learning based on inviters features, which includes average number of invitations sent by inviters, average number of activated users by inviters, and average success ratios of inviters; and (ii) learning based on invitee features, which includes total number of invitations received by invitee, average number of invitations sent by inviters to invitee, elapsed time for receiving invitations, reception burstiness, and gift variety.

Figure 7 plots the performance of the models based on invitee and inviters features, respectively, in terms of precision, TPR (or recall), FPR, ACC, and AUC. 'All' considers both invitee and inviters features. Note that Table III summarizes the performance of the three models. As shown in Figure 7 and Table III, the performance of the model using only inviters features is much better than the performance of the model based on only invitee features. The precision and AUC of the model using only inviters features are better than those of the model using only invitee features. Since the model using only invitee features cannot identify almost any of the adoptions correctly, its recall is almost 0. However, the model based on inviters features show higher recall (34.8%), and the recall increases to 41.3% for the model that uses both invitee and inviters features. Although the model based solely on inviters features is substantially effective

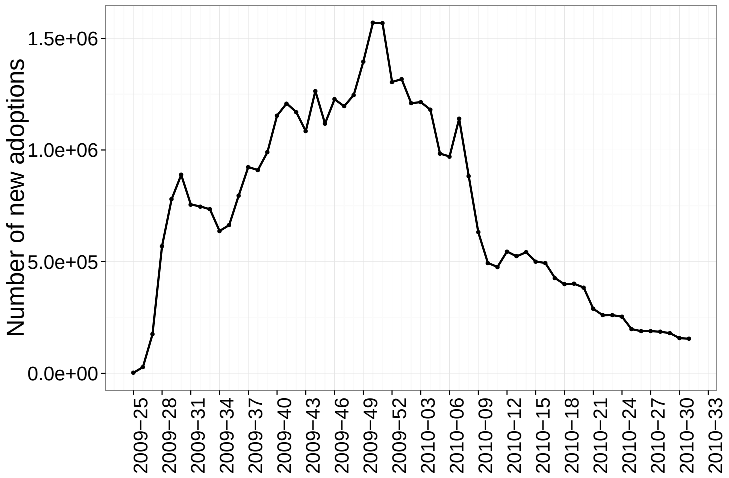


Fig. 8. Total number of new adoptions for each week on iHeart. We notice three phases in iHeart: i) growth (up to 2009-50), ii) peak (2009-50 to 2010-01), and iii) decline (2010-1 onwards). Sharp dips/spikes in application adoptions are due to holidays and special real-world events. From left to right, these dips are due to: Halloween, Thanksgiving, Christmas/New Year, and Valentine's Day.

in predicting application adoption, its performance can be further improved by adding the invitee features as well. **Note that the precision, TPR, FPR, and ACC of the model to predict whether an individual user *does not adopt* the application, i.e., the major class in our case, based on invitee and inviters features are 0.856, 0.949, 0.589, and 0.836, respectively. The macro-AUC is 0.857. The precision and TPR values are significantly high while the FPR value is relatively high, meaning that the model identifies the non-adoption cases very well, but it tends to over-detect them. Since our focus is 'identifying adoptions of individual users' and the non-adoption models are prone to over-detect the adoptions, we report the results of adoption models (i.e., the models to detect the minor class in our case) hereafter.**

Table III. Summary of performance of three models based on learning from (i) invitee features, (ii) inviters features, and (iii) both invitee and inviters features ('All').

Phases	Prec	TPR	FPR	ACC	AUC
Invitee features	0.361	0.003	0.001	0.788	0.666
Inviters features	0.683	0.348	0.043	0.828	0.822
All	0.688	0.413	0.050	0.836	0.857

In summary, the inviters features are more important than the invitee features for predicting application adoption, and building a model based on inviters features alone can achieve good performance to predict application adoptions. In addition, using both invitee and inviters features in the model further improves the performance over using only the inviters features.

5.3.2. Application Evolution Phases. We next investigate the model performance in predicting the adoption of the users observed during different application evolution phases (e.g., growth, peak, and decline). To this end, we first plot the total number of new adoptions of iHeart for each week in Figure 8. We observe three phases in iHeart: i) growth (up to 2009-50th), ii) peak (2009-50th to 2010-01st), and iii) decline (2010-01st onwards). We observe that percentages of the number of users who adopt the application during three phases are different. During the first phase (i.e., growth) of iHeart, we find over 25% of the iHeart population adopts the application, which explains why iHeart became popular at the end of year 2009. However, only 15% of the population during the peak and 13% of the population during the declining phase adopt iHeart. Note that over 60% of iHeart users appear during the growth of iHeart.

We now build a model by training from all the phases of iHeart evolution, and test on different phases to explore the robustness of our model. For the purpose of 10-fold cross validation, we divide the iHeart dataset randomly into 10 equal-size sets, and, in each fold, we choose one of them as a test set for testing and the other nine sets as training sets for learning the model. In each fold, the test set is further divided into three equal-sized ($|T|$) groups based on different phases of iHeart: (i) $T - growth$ is the set of users observed in the growing phase of iHeart, (ii) $T - peak$ is the set of users observed in the peak phase of iHeart, and (iii) $T - decline$ is the set of users observed in the declining phase of iHeart. We also create $T - iHeart$, which is a $|T|$-size test set of users selected randomly from all of the users in the test set who appeared in any phase for comparison purposes.

Table IV. Performance of our model on different application phases: (i) $T - growth$ is the set of users observed in the growth phase of iHeart, (ii) $T - peak$ is the set of users observed in the peak phase of iHeart, (iii) $T - decline$ is the set of users observed in the declining phase of iHeart, and (iv) $T - iHeart$ is the set of users observed in any phase of iHeart. Note that the model is trained by iHeart users sampled from all three phases of the application evolution.

Phases	Prec	TPR	FPR	ACC	AUC
$T - growth$	0.733	0.460	0.058	0.819	0.870
$T - peak$	0.545	0.281	0.042	0.856	0.809
$T - decline$	0.507	0.274	0.041	0.866	0.811
$T - iHeart$	0.686	0.414	0.051	0.836	0.858

Table IV shows the performances of our model on four different test sets: (i) $T - growth$, (ii) $T - peak$, (iii) $T - decline$, and (iv) $T - iHeart$. We find that our model performs more precisely with higher recall when testing on $T - growth$ than testing on $T - peak$ and $T - decline$. The precision of our model on $T - growth$ is over 73%, which means that 73% of the adoption identification during $T - growth$ are correct. The AUC of our model on $T - growth$ reaches to 0.87, which shows that our model is substantially effective in identifying adoptions for the users observed during the application growth. The precision and recall of our model on $T - peak$, $T - decline$, and $T - iHeart$ are lower than those on $T - growth$. Interestingly, AUC for $T - peak$ and $T - decline$ is consistently as high as 81%, which indicates that the model performs well during the peak and decline phase but not as good as when testing on $T - growth$. The performances on $T - growth$ are even slightly higher than those on $T - iHeart$, which means our model is more effective in predicting application adoptions for the users in $T - growth$.

In summary, our learning based model exhibits high AUCs (over 80%) across any evolution phases of iHeart, which confirms its general effectiveness in predicting user adoptions. We show that the model performs better during the growth of iHeart, where over 25% of the iHeart population adopt the application, than other phases in the evolution.

Table V. Performance of the models based on learning from other users in similar gifting applications. We learn the features of other users from other applications (i.e., Hugged and/or iSmile). We also collectively learn our model based on iHeart and other applications, Hugged and/or iSmile. We test different models on the same set of iHeart users. For comparison purposes, in the first row, we show the performances of the model learned from iHeart alone.

Applications	Prec	TPR	FPR	ACC	AUC
iHeart	0.688	0.413	0.050	0.836	0.857
Hugged	0.750	0.246	0.022	0.823	0.851
iSmile	0.781	0.194	0.015	0.818	0.857
Hugged + iSmile	0.768	0.217	0.018	0.820	0.855
iHeart + Hugged	0.714	0.337	0.036	0.831	0.858
iHeart + iSmile	0.742	0.302	0.028	0.830	0.858
iHeart + Hugged + iSmile	0.744	0.286	0.027	0.828	0.858

5.3.3. Learning Activity Information of Other Users from Other Gifting Applications. We next explore whether or not learning from similar type (i.e., gifting genre) of applications can help to predict user adoptions of iHeart. To this end, we build multiple learning models: learn activities from target application (iHeart) as well as two other applications (i.e., Hugged and/or iSmile). As a result, we have seven learning models in Table V. We test the learning models on iHeart users from any phase of the evolution.

Table V shows the performance results of learning from different applications and their combinations. We first examine how well a model learned from another application can predict the user adoptions of the target application. The second and third rows in Table V show the performance results of following two models: (i) learning from Hugged and testing on iHeart and (ii) learning from iSmile and testing on iHeart. Interestingly, learning from other similar application shows higher precision compared to learning from the target application (iHeart). Although their recall is much lower than that of learning form iHeart, their performance is almost as good as iHeart with respect to AUC. We also learn from both Hugged and iSmile together and test on iHeart, which shows similar performances compared to learning from either application alone.

We next investigate whether or not learning from the target application as well as other applications together helps to increase performance. To this end, we learn user activities from Hugged and/or iSmile, on top of iHeart. The last three rows in Table V show the performances on iHeart + Hugged, iHeart + iSmile, and iHeart + Hugged + iSmile, respectively. We observe that their performances are mostly similar to 'iHeart', which learns user activities from iHeart alone. The AUCs are almost similar, which indicate that learning and testing only from the target application is enough to predict the application adoption. Although performances of different models are almost similar in terms of the AUC, using the target application (iHeart) alone for learning could be more desirable since it provides balanced precision and recall. Hence, blindly applying learning on a dataset from one application and testing on a dataset of a different application is not particularly helpful because the overlap of the underlying user population may be small, and there may be inherent user bias. We explore this in more detail in Section 5.3.4 when we focus on 'common' user-base that are active in multiple applications.

5.3.4. Learning Additional Activity Information of Individual Users from Other Gifting Applications. **We finally investigate whether or not additional activity information (i.e., both inviter and invitee features described in Section 5.3.1) of individual users in other similar type applications are effective to improve the performance of the adoption prediction model.** For this, we learn and test the following

Table VI. Model performances based on learning from individual user activity information in other applications. We learn and test models only on the common users who participated in all three applications, Hugged, iSmile, and iHeart. "iHeart only" learns eight user features in iHeart only. "iHeart + adoption information in other applications" learns the eight user features in iHeart of each common user and two more features that indicate whether or not a targeted user adopts Hugged and/or iSmile, respectively. "iHeart + adoption & activity information in other applications" learns the eight user features in iHeart of each common user, two features indicating whether or not each of the targeted users adopt Hugged and iSmile, and sixteen user activity features in Hugged and iSmile. We test these models to predict the user adoptions in any phase of iHeart given the user participated in all three applications.

Learning Models	Prec	TPR	FPR	ACC	AUC
iHeart only	0.772	0.639	0.061	0.866	0.919
iHeart + adoption information in other applications	0.794	0.666	0.055	0.877	0.928
iHeart + adoption & activity information in other applications	**0.834**	**0.706**	**0.045**	**0.894**	**0.939**

models on *common users*, who were observed in all three applications, Hugged, iSmile, and iHeart. Note that the common users are the ones who have received at least one AR in each of the three applications. As mentioned in Section 3, all three applications are shared by around 16.5 M users among the total 221 M users. Table VI shows the performance of the three models: "iHeart only", "iHeart + adoption information in other applications", and "iHeart + adoption & activity information in other applications". As a baseline, in "iHeart only", we first learn from a training set of iHeart related to a subset of common users, and then test the model on a test set of iHeart related to another subset of common users. Interestingly, performance metrics of "iHeart only" are much higher than those of the model learned and tested on all iHeart users (Table V first row); The precision and recall of "iHeart only" are 77% and 64%, respectively. The AUC of "iHeart only" reaches to 0.919, implying "iHeart only" is a substantially effective model in predicting application adoptions of common users. This is likely because the common users in all three applications are more loyal to using gifting applications, and thus they show consistency in their activities and adoption behaviors.

To explore how additional activity information of an individual user in other applications are effective in predicting the adoption of target application, we learn not only user activities from iHeart, but also information that indicate whether or not he/she adopts the Hugged and iSmile in "iHeart + adoption information in other applications". That is, "iHeart + adoption information in other applications" learns total ten features: eight activity features from iHeart, and two binary (0 or 1) adoption features that indicate whether or not a user adopts Hugged and/or iSmile. In "iHeart + adoption & activity information in other applications", we learn not only user features in Heart and two binary indicators whether or not a user adopts Hugged and/or iSmile, but also user activity information in Hugged and iSmile. In other words, "iHeart + adoption & activity information in other applications" learns total twenty six features: eight activity features from Hugged, eight activity features from iSmile, eight activity features from iHeart, and two binary indicator features whether or not a user adopts Hugged and iSmile, respectively.

As shown in Table VI, the precision and recall of "iHeart + adoption information in other applications" are 79.4% and 66.6%, which is higher than those of "iHeart only". This indicates that including adoption information in other applications can enhance the prediction model. Also, AUC of this model is 92.8%, which is higher than that of "iHeart only". Finally, the model based on user activity features in iHeart and adoption information in Hugged and iSmile, as well as user activity information in Hugged and iSmile results in higher performances over all the previous models. The AUC of the model has improved (93.8%), and also we notice significant improvements of precision and recall (83.4% and 70.6%, respectively), meaning that the "iHeart + adoption & activity information in other applications" is most effective in predicting application adoptions for the *common users* than the previous models. **Note that the**

precision, TPR, FPR, ACC, and AUC of the model to predict whether an individual user *does not adopt* the application, i.e., the major class in our case, using the above features are 91.0%, 95.4$, 29.3%, 89.4%, and 93.9%, respectively, which signifies that those features are also very effective in predicting non-adoptions of the application for the common users. In conclusion, using additional information of individual users from other similar type of applications can significantly improve the predictive power of the application adoption.

6. CHARACTERIZING CASCADING TREES OF IHEART

In this section, we describe how new applications are adopted and propagated over OSN platform, using a Facebook third-party gifting application as a case study. To this end, we characterize the cascades of iHeart adoptions from a graph-theoretical perspective, and investigate what factors drive the process of application adoptions.

6.1. Definition of cascading tree

To analyze how a gifting application is adopted (and propagated) in Facebook, we define a cascading tree as a directed graph, $G = (V, E)$, where V is the set of users and E is the set of invitations to adopt, based on the Krackhardt's hierarchical tree model [Hanneman and Riddle 2005]. That is, if user j adopts an application from user i's invitation, there exists an edge $E(V_i, V_j)$ from user V_i to user V_j. We refer to this event as *adoption* of an application. After a user adopts the application, he/she will in turn send out invitations to his/her friends, thereby recruiting more users. A sequence of adoptions is denoted by a *cascade*. **Note that a user must install the application before he/she can send an item through the application. Hence, all the senders (i.e., users whose sending activities are recorded in our data) must have installed (or adopted) our application. The caveat is that our data does not allow us to verify when or if a user who receives an invitation installs an application, until that the user uses the application to send an invitation (i.e., becomes a sender). Hence, the leaf nodes in our cascading trees are users who receive one or more invitations but do not send any invitations. The leaf nodes are users who do not contribute to propagating the adoption cascades further.**

Figure 9 illustrates an example of a cascading tree. The circles represent users who belong to the cascading tree G. A cascading tree is initiated by a seed user (say, v_0 in this example) who already installed the application (before receiving any invitations) and have sent invitations. The *max depth* in a cascading tree is defined as the hop count from the root (i.e., seed) to the farthest leaf node in the tree. The *max width* refers to the number of nodes in the widest generation in the tree. In Figure 9, the size, max depth, and max width of the cascading tree G are 12, 3, and 6, respectively. Since we focus on adoption process of applications, we consider only the cascading trees that have at least one invitation-induced adoption, i.e., whose sizes are at least 2.

Since a not-yet recruited user can receive multiple invitations (from multiple inviters), his or her set of potential parents is a group of senders from which the user received invitations before his/her adoption. Note that a parent who is actually responsible for the adoption of a particular user is not known in our dataset. We seek to enforce a relationship between an actual inviter and the activated user by identifying *a single parent* for each activated user. To infer such a parent who is responsible for the adoption of a particular user, we first calculate each sender's (or inviter's) success ratio, τ, which was introduced in Eq. 1. Since the sender whose τ is the highest is the most probable sender (or inviter) responsible for the adoption of a given user, u, we finally select that sender as the parent of u in a same cascading

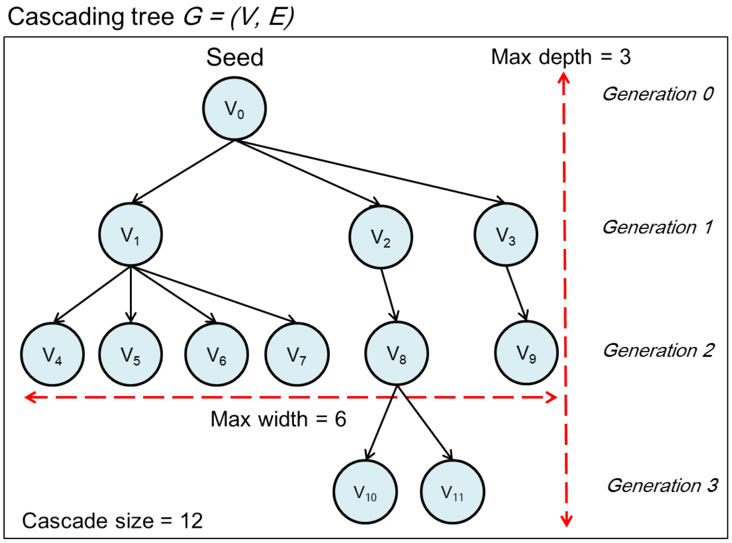


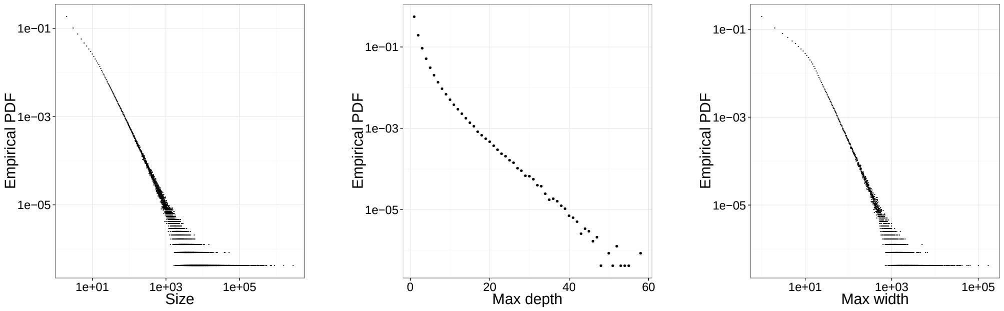
Fig. 9. Definition of cascading tree.

tree. Note that our previous work *et al.* [Rahman et al. 2014] evaluated four other methods to infer the potential parent: i: i) the node who sent the invitation to the particular user/receiver for the first time before the receiver adopts the application, ii) the last node who sent the invitation, iii) the node who sent the most number of invitations among all the senders, and iv) a node randomly selected among the parents. We find that our method of selecting a parent based on the success ratio shows similar distributions of cascade sizes compared to the four methods in [Rahman et al. 2014], but we adopt the method of identifying potential parent based on the success ratio since we are studying the application adoption process and are interested in identifying the most effective sender (or inviter) who succeeds in recruiting the most number of new users.

6.2. Properties of cascading trees

We first explore the application adoption process by examining the cascading tree properties that were defined in Section 3, namely the cascade size, max depth, and max width. Then we investigate the properties of cascading tree evolutions during their lifetimes as well as the roles of users based on their contributions to the growth of the cascade. Recall that the size, max depth, and max width of a cascading tree are the number of recruited users, the maximum distance from the seed to the unadopted users (leaf nodes), and the number of users in the widest generation of that tree, respectively.

6.2.1. Properties of tree structure. Figure 10 illustrates the structural properties of cascading trees in terms of size, max depth, and max width. In Figure 10(a), we observe that 18% of the cascading trees are of size 2, which indicates that seeds of a substantial portion of the cascading trees propagate the application to only one child. Moreover, 93% of the cascading trees are of size less than 100. Note that the maximum and average sizes of the cascading trees are 2,850,149 and 79.10, respectively. The top 10% of the cascading trees (ranked by size) have at least 70 nodes. We refer to these as the *'large'* cascading trees. We also observe that only a small fraction of the cascading trees



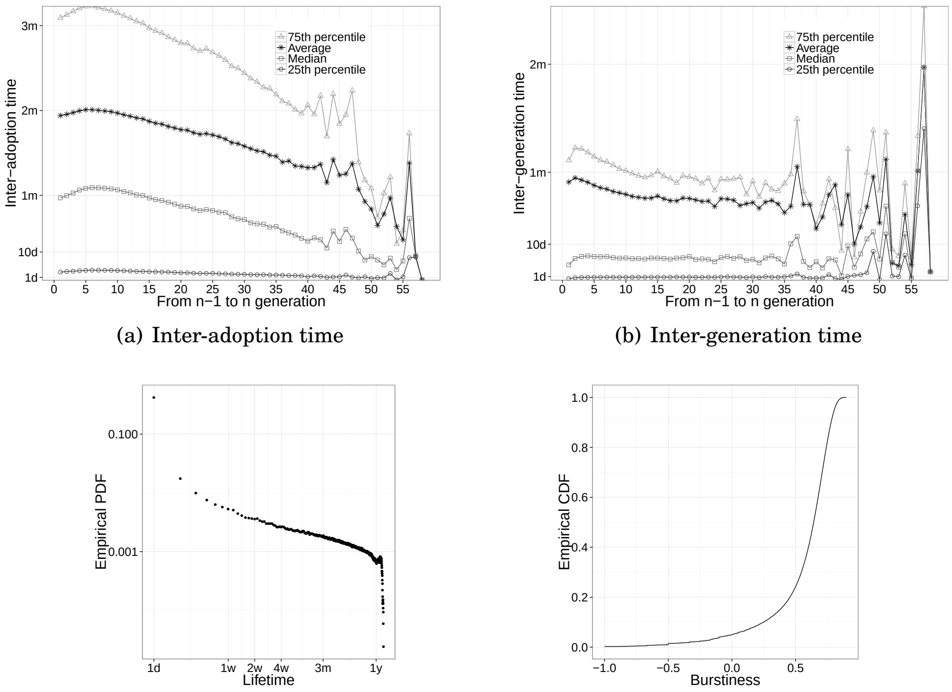
(a) Cascade size
(b) Cascade max depth
(c) Cascade max width

Fig. 10. Structural properties of iHeart cascading trees.

grow deep in Figure 10(b). The depths of 75% of the cascading trees are 1 or 2, while 7% of the cascading trees span deeper than 5 generations. The maximum and average depth of the cascading trees are 58 and 2.26, respectively. We notice that the max depth of the cascading trees in iHeart is higher than that in Twitter (11) [Kwak et al. 2010] or Pinterest (35) [Han et al. 2014], which implies that the propagation of OSN-based applications is deeper than that of other information such as tweets or pins. When we look at the cascade max width in Figure 10(c), we find that most of the cascading trees are narrow; 80% of the cascading trees have less than 10 max widths. The maximum and average width of the cascading trees are 171,033 and 17.64, respectively. Note that the largest max depth and max width are 58 and 171,033; they are four-orders of magnitude different, which means cascading trees in iHeart are generally much wider instead of being deeper.

6.2.2. Properties of cascade evolution. To understand the growth of application adoption process, we investigate i) inter-adoption time, ii) inter-generation time, iii) lifetime, and iv) burstiness of a cascading tree, as explained below. We believe such observations can provide an implication on how cascading trees evolve (from a temporal perspective) during the adoption process.

Inter-adoption time is the time difference between the adoption time of a parent and each of its children in a cascading tree. Let's say, a node u and its child v in a cascading tree adopted the application at time t_u and t_v, respectively, where $t_u < t_v$. Then, the inter-adoption time i_{uv} between u and v is $t_v - t_u$. We calculate inter-adoption times for all the inviter-invitee pairs in a cascading tree. That is, a set of inter-adoption times IA of a cascading-tree is $\{i_{01}, i_{02}, \ldots i_{uv}\}$. We find that 83.28% of all the inter-adoption times are longer than a day, and more than 23% of them are longer than 100 days, which implies that the propagation speed of OSN-based applications (i.e., iHeart) is slower than that of other information such as tweets in Twitter [Kwak et al. 2010] and pins in Pinterest [Han et al. 2014]. To investigate the inter-adoption times between two consecutive generations, we show the 25th percentile, median, 75th percentile, and average of inter-adoption times from $n-1$ hop to n hop in Figure 11(a). We find that there is a large variation in the inter-adoption time at every level of the tree. Between the seeds and first generation nodes, inter-adoption time spans from minimum of 2 seconds to maximum of 227 days. This dynamic range as well as the median decreases for later generations, i.e., iHeart application propagates from one user to another at a slightly faster speed for late adopters compared to early users. This could be due to the fact that the general maturity and popularity of the application increased in the later stage. Overall, we observe that the median value decreases from 29-34 days (about a



(a) Inter-adoption time

(b) Inter-generation time

(c) Lifetime

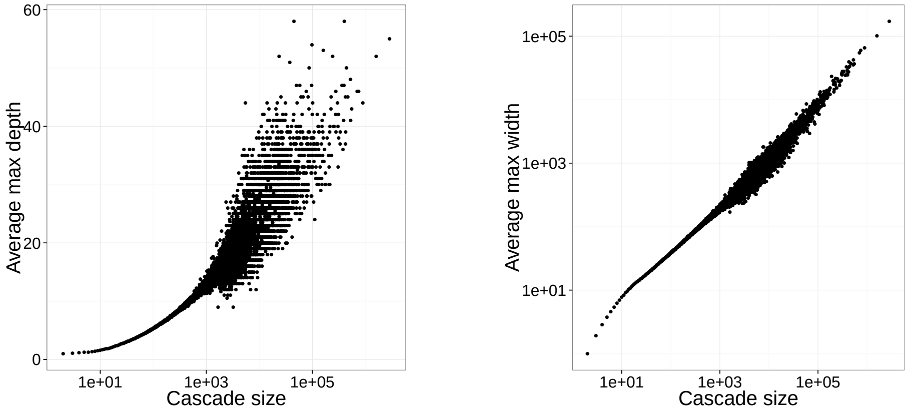
(d) Burstiness

Fig. 11. Evolutionary properties of iHeart cascading trees.

month to propagate) near the root of the cascading tree to about 13 days at generation 40.

Inter-generation time is the time difference between the first adoption time in two consecutive generations. Say, two consecutive generations g_{n-1} and g_n of a cascading tree include $n(g_{n-1})$ and $n(g_n)$ nodes with adoption times $T(g_{n-1}) = \{t_{g_{n-1},1}, t_{g_{n-1},2}, \ldots, t_{g_{n-1},n(g_{n-1})}\}$ $T(g_n) = \{t_{g_n,1}, t_{g_n,2}, \ldots, t_{g_n,n(g_n)}\}$, respectively. Then, the inter-generation time is defined as $IG(n-1,n) = min(T(g_n)) - min(T(g_{n-1}))$. We plot the inter-generation time in Figure 11(b). As shown in Figure 11(b), the median inter-generation times are below 7 days until the generation 35. After the generation 35, the inter-generation times fluctuate widely. In other words, for 50% of the time, the cascading tree takes about a week to add another level (generation). Note that the average and median of the inter-generation times between the seed and the first generation are 27.2 days and 4.3 days, respectively, which implies that the propagation speed of iHeart across different generations is much slower than that of tweets in Twitter [Kwak et al. 2010] or pins in Pinterest [Han et al. 2014].

Lifetime of a cascading tree is the duration between the first time sending an invitation by a seed and the last time any node in the cascading tree receives an invitation. We show the distribution of lifetimes of all the cascading trees in Figure 11(c), which can provide implication on how long cascading trees evolve during the adoption process. We observe that many cascading trees show long lifetimes, which signifies that they evolve for a long time during the adoption process. For instance, 3% of the cascading trees have lifetime of more than a year,



(a) Avg. max depth

(b) Avg. max width

Fig. 12. Avg. max depth and width corresponding to different cascade sizes.

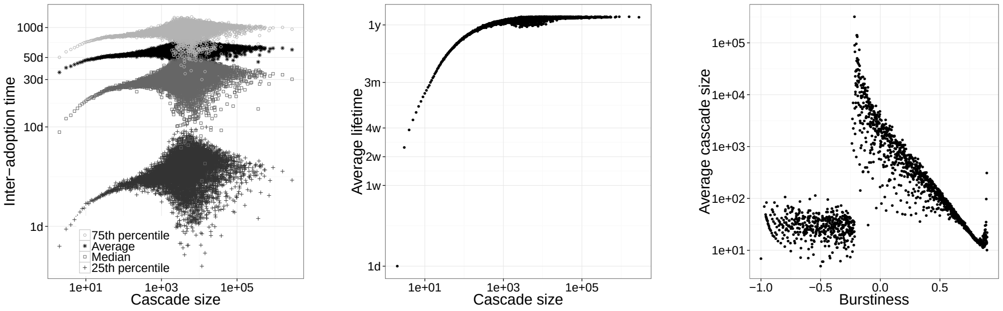
and 58% of the cascading trees have lifetime of longer than a day. The average lifetime of all the cascading trees is 81.1 days.

Burstiness of tree evolution measures the regularity in the evolution process of the cascading trees. We can calculate the burstiness parameter B for the tree evolution (i.e., how regularly adoptions emerged in a cascading tree) using Eq. 5. Instead of the set of number of invitations in each time slot, we re-define X in Eq. 5 that represents the set of number of nodes in a cascading tree in each time slot. That is, $x_t \in X$ where $X = \{x_1, x_2, x_3, \ldots x_t\}$ indicates that x_t nodes emerged during the given time slot t. We take time slot as 1 day. Figure 11(d) shows the CDF of burstiness B of tree evolution for each cascading tree. As shown in Figure 11(d), B of 5% of the cascading trees are below 0, which means they grow from regularly (closer to -1) to randomly (closer to 0) during their lifetimes. However, 80% cascading trees have B value of over 0.45, and the maximum B is 0.897, meaning that they exhibit very skewed (or bursty) growth; e.g., their growths happen predominantly during the beginning of their lifetimes.

6.3. What are the distinct features and driving factors for successful adoption?

We next explore what factors drive the growth of the cascading trees and whether there are distinctive features associated with different cascade sizes. To this end, we investigate i) structural factors (e.g, max depth or width of a cascading tree), ii) evolutionary factors (e.g., burstiness), and iii) roles of nodes (e.g., contribution of a seed to a cascading tree).

6.3.1. Structural properties of cascading trees. We first show the average max depth and max width corresponding to the different cascade sizes in Figure 12. Note that we report the average max depth and width values for each cascade size for the visualization purpose, because there are more than 2.3 M cascading trees in our dataset, plotting all the cascading trees could not be distinguishable. We only find a weak positive correlation between the cascade size and the average depth of the cascades (Pearson correlation coefficient is 0.365, $p - value < 0.001$ **with 95% confidence interval**); not all big cascading trees span deep. In particular, the cascading trees whose sizes are bigger than 1000 have at least 9 generations on average. On the other hand, the average max width shows a significantly high correlation with the size; Pearson correlation coefficient is 0.985 ($p - value < 0.001$ **with 95% confidence interval)**. This suggests that big cascading trees are wide but small cascading trees are narrow on average.

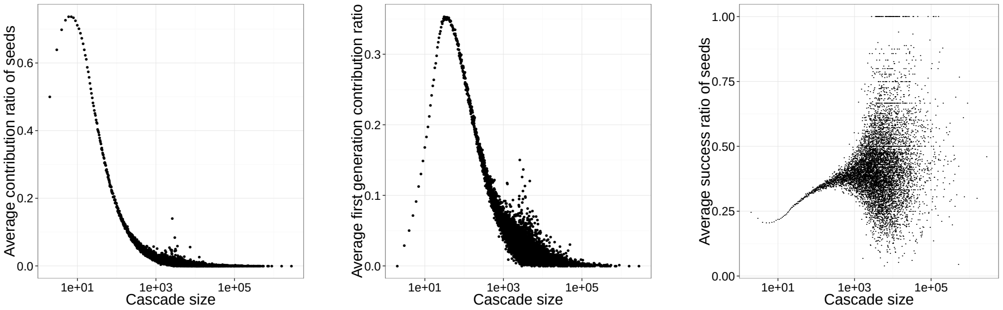


(a) Cascade size vs. inter-adoption time

(b) Cascade size vs. avg. lifetime

(c) Burstiness vs. cascade size

Fig. 13. Ultimate cascade sizes and evolutionary properties of cascading trees: (a) inter-adoption time, (b) lifetime, and (c) burstiness.



(a) Cascade size vs. avg. contribution ratio of seeds

(b) Cascade size vs. avg. first generation contribution ratio

(c) Cascade size vs. avg. success ratio of seeds

Fig. 14. Exploring the role of early adopters (seed and first-generation nodes) in the cascade growth.

6.3.2. Evolutionary properties of cascading trees. We investigate how the following three evolutionary properties of cascading trees vary with respect to cascade sizes in Figure 13: i) inter-adoption time, ii) lifetime, and iii) busrtiness. Since the inter-generation time shows similar patterns with the inter-adoption time, we do not include the results for the inter-generation time due to space limitation. First, we plot the average, 25th percentile, median, and 75th percentile of the inter-adoption times of the cascading trees based on the cascade sizes in Figure 13(a). We find that the inter-adoption time and the cascade size have a positive correlation until the cascade size is 1000. Surprisingly, the dynamic range of the average inter-adoption time changes sharply, ranging from 26 days to 77 days, after the cascade size reaches 1000. However, there is no clear correlation between the inter-adoption time (or inter-generation time) and the ultimate cascade size. Figure 13(b) shows the correlation between the average lifetime and the cascade size. We find that smaller cascading trees have shorter lifetimes on average compared to the big ones. The average lifetime increases as the cascade size increases until the cascade size is close to 700, but it converges to around 1 year after the cascade size reaches 700. We next investigate the correlation between the average cascade size and the burstiness values of cascading trees in Figure 13(c). Interestingly, we observe two clusters in Figure 13(c): i) burstiness range between -1 to -0.22 and ii) burstiness range between -0.22 to 1. If a cascading tree has a uniform burstiness (closer to -1), its size tends to be less than 100. We observe that there is a negative correlation between the average cascade size and the burstiness. That is, the size of a cascading tree decreases as the burstiness increases. This implies that skewed growing patterns, such as initial spikes followed by inactivity, fails to sustain the population growth, leading to smaller cascades. We find that large cascading trees tend to have a burstiness value around zero, which implies the adoption process for large cascades tend to follow a random (but sustainable) growth patterns over time.

6.3.3. Roles of initial adopters. Lastly, we explore whether the seeds and early adopters play a significant role in driving the growth of the cascading trees in Figure 14. We analyze: i) the contribution ratio of the seed (i.e., founder of the cascading tree), ii) the contribution ratio of the nodes in the first generation (which may be the co-founders of the cascading tree), and iii) the success ratio (defined in Section 3) of the seed. Here, the contribution ratio of a node is defined as the ratio of number of its children to the total population of the cascading tree. We observe that the average contribution ratio of the seeds decreases as the cascade size increases, as shown in Figure 14(a). This implies that the seeds play a diminishing role in growing the user base as the cascade size increases. In fact, for large cascade sizes (70 or above), the average contribution ratio of the seeds are generally small (75th percentile is 0.012), i.e., the seeds are only responsible for recruiting a tiny fraction of the total population in the large cascades. The initial expansion of user base due to the seed is generally not a strong indicator of the ultimate size of the cascading tree. Figure 14(b) plots the average contribution ratio of the first generation nodes, which initially increases with respect to cascade size until 31 and then it decreases as cascade size increases. On the other hand, Figure 14(c) shows that the average success ratio of the seeds is somewhat positively correlated with the cascade size up till around 1000. Note that the Pearson correlation coefficient till the cascade size of 1000 is 0.63 **($p - value < 0.001$ with 95% confidence interval)** while that beyond the cascade size of 1000 is 0.04 **($p - value < 0.001$ with 95% confidence interval)**, which implies that the success ratio of seeds alone may not be only the important factor for generating large cascading trees.

7. PREDICTION ON CASCADE GROWTH

Our measurement-based characterization of iHeart adoption process suggests that there exist a set of distinctive features (e.g., structural or evolutionary properties of cascades and success ratio of their seed nodes) that can be combined to predict the final cascade size. In this section, we seek to answer the following questions: (1) How can we predict the growth of the cascading process of an application? (2) How long (e.g., 2 weeks or 1 month) do we need to observe the initial growth of a cascade? and (3) What features can we exploit to forecast the cascade growth with an acceptable precision? Leveraging insights gained from our measurement study, we propose a learning-based prediction model to identify *large-scale cascades of application adoption* by observing the relevant features of the cascades in the initial adoption process.

7.1. Prediction as a learning problem

7.1.1. Problem definition. Our goal is to identify large cascading trees whose sizes are 70 or above; this accounts for the top 10% of the cascading trees in terms of size. **The top 10% of the cascading trees are responsible for 86.5% of the newly recruited**

Table VII. Feature descriptions.

Categories	Features	Descriptions
Structural	$d_w(i)$	Max depth of the cascade i at the end of the observation window w
	$w_w(i)$	Max width of the cascade i at the end of the observation window w
Evolutionary	$b_w(i)$	Daily burstiness of the cascade i during the given observation window w
	$g_w(i)$	Growth rate of the cascade i during the given observation window w
Early adopters	$c_{s,w}(i)$	Contribution ratio of the seed s of the cascade i during the given observation window w
	$c_{f,w}(i)$	Contribution ratio of the nodes f in the first generation of the cascade i during the given observation window w
	$s_{s,w}(i)$	Success ratio of the seed s of the cascade i during the given observation window w

users in iHeart. We cast this as a learning problem, where we observe the *initial growth patterns* of a cascade and forecast whether an adoption will reach a certain cascade size (i.e., 70) in the future. **Note that our prediction study assumes a practical situation: at the time of observation (e.g., 1-week, 2-week), we do not know when the growth of the cascade ends. Thus, our prediction model only considers the initial features of a cascade (which is observable), and forecasts whether the cascade belong to the top 10% of all the cascades.**

7.1.2. Observation window. To investigate the sensitivity of our prediction result with respect to the length of the learning phase, we vary the observation window w from one week to four weeks. For example, if the w is two, we observe the initial characteristics of a cascade until the second week. At the end of each observation window, we extract a set of features (which will be described later) associated with the cascading trees that capture the initial adoption process of iHeart. Note that we do not assume the growth of a cascade has stopped at the end of the observation window. Instead, we predict whether the observed growth will remain small (i.e., with cascade size less than 70) or grow large in the future.

7.1.3. Factors driving cascade growth. To forecast the cascade growth, we observe a set of distinctive features associated with the adoption process during a given observation window. The candidate features described in Table VII are selected based on their demonstrated predictive power in our measurement study, i.e., how well these features can help to identify the large cascades. By observing the initial growth patterns of the given cascade, which include the structural and evolutionary patterns, and early adopter's characteristics, we predict the large application cascades in the future. Note that we do not consider inter-adoption time and inter-generation time here as distinctive features for learning because of the lack of correlation between these features and cascade size. We also do not consider lifetime further since it may not be a proper measure in the initial stage; instead, we use the growth rate of the cascade i, $g_w(i)$, as a distinctive feature for evolutionary properties. $g_w(i)$ can be calculated as the cascade size of i (during the observation period) divided by w. Note that $c_{s,w}(i)$ and $c_{f,w}(i)$ are calculated as the number of children of s and f in i divided by the initial size of cascade i during w, respectively.

We first identify the specific features that contribute most towards predicting large cascades, and then we use them to create the model. For this purpose, *Chi-squared* (χ^2) statistic evaluation [Liu and Setiono 1995] (which was explained in Section 5) is applied to all of the above mentioned features. Note that χ^2 statistic is used to evaluate the 'distance' between the distribution of the large and small cascades for an attribute. The bigger the value, the more effective is the feature in identifying the large cascades. **All the $p-values$ are lower than 0.001 with 95% confidence interval).** As shown in Table VIII, the most important feature is g_w, which means the growth rate of a cascade is a good indicator of the ultimate cascade size. Max depth (d_w) and max width (w_w) are the next two important features for $w > 1$. The contribution ratio of the seeds is the most important feature among the three properties of early adopters. Both of the contribution ratio of the first generation users and the success ratio of the seeds have similar predictive power with respect to large cascade. Interestingly, the χ^2 values of the properties of early adopters decrease (i.e., $c_{s,w}$, $c_{f,w}$, and $s_{s,w}$) as w increases, meaning that early adopters (seeds and their recruited users) play a diminishing role as time goes on. The burstiness factor has the lowest score and, if used alone, is the least effective in predicting the large cascades.

Table VIII. Feature importance with normalized χ^2.

Features	$w = 1$	$w = 2$	$w = 3$	$w = 4$
g_w	1.000	1.000	1.000	1.000
d_w	0.910	0.907	0.900	0.892
w_w	0.695	0.723	0.738	0.750
$c_{s,w}$	0.740	0.708	0.688	0.673
$c_{f,w}$	0.615	0.551	0.514	0.486
$s_{s,w}$	0.673	0.542	0.475	0.432
b_w	0.598	0.490	0.434	0.397

7.2. Performance

We build logistic regression models based on the above features measured/estimated during the observation window from the first to fourth week of every cascade: (i) *"Structural"* considers two structural properties (i.e., d_w and w_w), (ii) *"Evolutionary"* considers two evolutionary properties (i.e., b_w and g_w), (iii) *"Early adopters"* considers three properties of initial adopters (i.e., $c_{s,w}$, $c_{f,w}$, and $s_{s,w}$), and (iv) *"All"* collectively considers all the features. We used various classifiers including SVM and decision trees, but we only report the performance of the logistic regression classifier since it performs similar or slightly better than other classifiers in most cases. We randomly set 2/3 of the 2,384,686 cascades (in iHeart) to train the model, and then we test our model with the other 1/3 of the cascades. Finally, we report the prediction precision, the true positive rate (TPR, or recall), the false positive rate (FPR), and area under the ROC curve (AUC) [roc 2006]. To calculate precision, TPR, and FPR, we use a *cutoff* probability as 0.5 to discretize continuous probabilities into binary decisions. Note that AUC resembles effectiveness of a prediction model; a perfect model has $AUC = 1$. See Section 5 for more explanations in detail.

Table IX summarizes the performance of our prediction model. As shown in Table IX, our model performs better as the observation window increases. We observe that the model based on the evolutionary properties outperforms other two models base on the structural properties and initial adopter properties, which confirms that evolutionary features are important predictors to identify the large cascades. On the other hand, the precision of the model based on the properties of early adopters is lower than other models; with four weeks of observations, the precision is only 0.694. Note that precision and TPR are already high for the observation windows $w = 1$ in our final model (i.e., "All"), meaning that one week of observation can provide a good prediction for identifying large cascades. After observing 4 weeks ($w = 4$), the precision and the recall of our final model is 82.8% and 58.7%, respectively. The AUC values of our final model for observation window $w = 4$ reaches to 0.960.

7.3. Universality of the proposed model

We finally explore whether our learning-based prediction model can be applied to predict the large cascades observed in other gifting applications. To this end, we use the

Table IX. Prediction performance on iHeart.

Features	measures	w=1	w=2	w=3	w=4
Structural	Precision	0.741	0.775	0.796	0.810
	TPR	0.344	0.438	0.492	0.535
	FPR	0.013	0.014	0.014	0.014
	AUC	0.860	0.890	0.907	0.921
Evolutionary	Precision	0.736	0.791	0.811	0.826
	TPR	0.332	0.433	0.494	0.539
	FPR	0.013	0.013	0.013	0.013
	AUC	0.901	0.926	0.938	0.947
Early adopters	Precision	0.431	0.613	0.667	0.694
	TPR	0.102	0.226	0.299	0.354
	FPR	0.015	0.016	0.017	0.017
	AUC	0.812	0.842	0.861	0.877
All	Precision	0.745	0.793	0.813	0.828
	TPR	0.415	0.495	0.548	0.587
	FPR	0.016	0.014	0.014	0.014
	AUC	0.929	0.945	0.954	0.960

Table X. Model performance on Hugged.

Features	measures	w=1	w=2	w=3	w=4
Structural	Precision	0.695	0.739	0.764	0.786
	TPR	0.252	0.356	0.417	0.460
	FPR	0.012	0.014	0.014	0.014
	AUC	0.817	0.858	0.880	0.895
Evolutionary	Precision	0.687	0.762	0.783	0.806
	TPR	0.238	0.368	0.435	0.483
	FPR	0.012	0.013	0.013	0.013
	AUC	0.874	0.913	0.930	0.941
Early adopters	Precision	0.599	0.626	0.648	0.666
	TPR	0.061	0.170	0.241	0.296
	FPR	0.005	0.011	0.014	0.016
	AUC	0.776	0.817	0.841	0.856
All	Precision	0.704	0.756	0.775	0.805
	TPR	0.316	0.425	0.482	0.527
	FPR	0.015	0.015	0.015	0.014
	AUC	0.907	0.932	0.945	0.953

dataset from *Hugged* application described in Section 3. We followed the same methodology outlined in Section 3 to extract the unique seeds and construct the different cascading trees for Hugged. For each cascade, we measure the above seven features over different observation windows, w = 1, 2, 3, and 4, respectively. Likewise, we define the top 10% (ranked based on size) of the cascades whose sizes are 99 or above as large ones. Again, we train our model with randomly chosen 2/3 of the 461,510 cascades, and test with 1/3 of the cascades in Hugged. The prediction results for Hugged are very similar to iHeart, as shown in Table X; the model based on the evolutionary properties of cascades outperforms the others. Our models achieve very high precision in identifying the large cascades also in Hugged. Note that the AUC value of our final model for observation windows $w = 4$ reaches to 0.953. Based on the prediction results for iHeart and Hugged, we believe our prediction model is generally applicable to gifting genre of OSN-based applications where new users are recruited primarily through user invitations.

8. CONCLUSION

This paper studied how user recommendation of OSN-based applications leads to their actual adoption and further propagation, which is the key to understanding and predicting the growth, popularity, and longevity of OSN-based applications. As a case study, we investigated which distinctive features of inviters and invitees are associated with the adoption of a popular OSN-based application, iHeart. We found that sending or receiving a large number of invitations does not necessarily help to recruit new users to iHeart, which may be due to the spamming phenomenon where users end up ignoring the invitations. We revealed that the average success ratio of inviters is the most important feature, meaning that capability or effectiveness of inviters has strong predictive power with respect to the application adoption. We also showed that the features related to invitation-receiving patterns are good indicators to predict application adoption. Based on the lessons learned from our measurement study, we built and evaluated learning-based models in predicting whether a user will adopt iHeart or not. We explored whether or not learning activity information of other users from similar gifting genre applications (i.e., Hugged or iSmile) can accurately predict the application adoption in iHeart. We found that learning past activities/behaviors of the same users from other similar applications (in our case, Hugged and iSmile) can enhance the predictive power of the adoption of the target application (iHeart), achieving high precision (83%). However, such transfer learning does not apply to different groups of users, i.e., learning activities of different group of users from other similar applications do not help in predicting the adoption behavior of users of the target application. This demonstrated that using behavioral information of each individual user in other similar applications is effective in identifying the OSN-based application adoption.

We next analyzed (1) what factors drive the cascading process of application adoptions, and (2) what are the good predictors of the ultimate cascade sizes. We discovered that the initial growth rate and shape of cascading trees (max depth and max width) are strong predictors of the final population size of the cascade. Based on the insights learned from our analyses, we proposed a prediction model for identifying the large-scale cascades based on observing the initial growth process. Results showed our proposed model can achieve high precision (over 80%) with four weeks of observation period. Our model also can achieve high precision when applied to Hugged, suggesting that our prediction model is generally applicable to gifting genre of OSN-based applications where the new users are recruited primarily through user invitations.

We believe our work on how user OSN-based applications are recommended, adopted, and propagated in the OSNs can provide important insight for OSN-based product stakeholders. This can be crucial for understanding and modeling the growth, popularity, and longevity of OSN-based applications or products. Also our proposed prediction models are useful for resource allocation of OSN-based product stakeholders, e.g., via targeted marketing or recommendation.

REFERENCES

2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (June 2006), 861–874.

2012. Facebook and Zynga to end close relationship. (2012). http://www.bbc.com/news/technology-20554441.

2013. AppData - Application Analytics for Facebook, iOS and Google Play. (2013). http://www.appdata.com.

Sinan Aral, Lev Muchnik, and Arun Sundararajan. 2009. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences* 106, 51 (2009), 21544–21549.

Sinan Aral and Dylan Walker. 2012. Identifying influential and susceptible members of social networks. *Science* 337, 6092 (2012), 337–341.

Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The Role of Social Networks in Information Diffusion. In *Proceedings of WWW*.

Jon Berry and Ed Keller. 2003. The Influentials: One American in Ten Tells the Other Nine How to Vote, Where to Eat, and What to Buy. Free Press. (2003).

Cha, Alan Mislove, and Krishna P. Gummadi. 2009. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of WWW*.

Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, and Krishna P. Gummadi. 2010. Measuring user influence in Twitter: The million follower fallacy. In *Proceedings of ICWSM*.

Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. 2014. Can Cascades Be Predicted?. In *Proceedings of WWW*.

Minas Gjoka, Michael Sirivianos, Athina Markopoulou, and Xiaowei Yang. 2008. Poking Facebook: Characterization of Osn Applications. In *Proceedings of WOSN*.

Sharad Goel, Duncan J. Watts, and Daniel G. Goldstein. 2012. The structure of online diffusion networks.. In *Proceedings of ACM EC*.

K. I. Goh and A. L. Barabási. 2008. Burstiness and memory in complex systems. *EPL (Europhysics Letters)* (2008), 48002+.

Sandra González-Bailón, Javier Borge-Holthoefer, Alejandro Rivero, and Yamir Moreno. 2011. The Dynamics of Protest Recruitment through an Online Network. *Scientific Reports* 1, 197 (2011).

Jinyoung Han, Daejin Choi, A-Young Choi, Jiwon Choi, Taejoong Chung, Ted "Taekyoung" Kwon, Jong-Youn Rha, and Chen-Nee Chuah. 2015. Sharing Topics in Pinterest: Understanding Content Creation and Diffusion Behaviors. In *Proceedings of ACM Conference on Online Social Networks (COSN)*.

Jinyoung Han, Daejin Choi, Byung-Gon Chun, Ted Taekyoung Kwon, Hyun-chul Kim, and Yanghee Choi. 2014. Collecting, Organizing, and Sharing Pins in Pinterest: Interest-driven or Social-driven?. In *Proceedings of ACM SIGMETRICS*.

Robert A. Hanneman and Mark Riddle. 2005. Introduction to social network methods. (2005). http://www.faculty.ucr.edu/~hanneman/.

Elizabeth A. Harris. 2013. Retailers Seek Partners in Social Networks. (2013). http://www.nytimes.com/2013/11/27/technology/retailers-seek-partners-in-social-networks.html?hp&_r=1&.

Jiwan Jeong and Sue Moon. 2014. Invite Your Friends and Get Rewards: Dynamics of Incentivized Friend Invitation in Kakaotalk Mobile Games. In *Proceedings of ACM Conference on Online Social Networks (COSN)*.

Zsolt Katona, Peter Pal Zubcsek, and Miklos Sarvary. 2011. Network effects and personal influences: The diffusion of an online social network. *Journal of Marketing Research* 48, 3 (2011), 425–443.

B. Kirman, Shaun Lawson, and C. Linehan. 2009. Gaming On and Off the Social Graph: The Social Structure of Facebook Games. In *Proceedings of International Conference on Computational Science and Engineering*.

Maksim Kitsak, Lazaros K. Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H. Eugene Stanley, and Hernan A. Makse. 2010. Identification of Influential Spreaders in Complex Networks. *Nature Physics* 6, 11 (2010), 888–893.

Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, and Andrey Kustarev. 2012. Prediction of Retweet Cascade Size over Time. In *Proceedings of ACM CIKM*.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a Social Network or a News Media?. In *Proceedings of WWW*.

K. Lerman and R. Ghosh. 2010. Information contagion: An empirical study of the spread of news on Digg and Twitter social networks. In *Proceedings of ICWSM*.

Yung-Ming Li, Han-Wen Hsiao, and Yi-Lin Lee. 2013. Recommending Social Network Applications via Social Filtering Mechanisms. *Information Sciences: an International Journal* 239 (2013), 18–30.

Albert M. Liebetrau. 1983. Measures of association. (1983). http://www.worldcat.org/isbn/9780585180977

Han Liu, Atif Nazir, Jinoo Joung, and Chen-Nee Chuah. 2013. Modeling/Predicting the Evolution Trend of OSN-based Applications. In *Proceedings of WWW*.

Huan Liu and Rudy Setiono. 1995. Chi2: Feature Selection and Discretization of Numeric Attributes. In *Proceedings of IEEE 24th International Conference on Tools with Artificial Intelligence*.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* (2001), 415–444.

Atif Nazir, Alex Waagen, Vikram S. Vijayaraghavan, Chen-Nee Chuah, Raissa M. D'Souza, and Balachander Krishnamurthy. 2012. Beyond friendship: modeling user activity graphs on social network-based gifting applications. In *Proceedings of ACM IMC*.

M. Rezaur Rahman, Jinyoung Han, and Chen-Nee Chuah. 2015. Unveiling the Adoption and Cascading Process of OSN-based Gifting Applications. In *Proceedings of IEEE INFOCOM*.

Mohammad Rezaur Rahman, Pierre-André Noël, Chen-Nee Chuah, Balachander Krishnamurthy, Raissa M. D'Souza, and S. Felix Wu. 2014. Peeking into the Invitation-based Adoption Process of OSN-based Applications. *Computer Communications Review* 44, 1 (2014), 20–27.

Tiago Rodrigues, Fabrício Benevenuto, Meeyoung Cha, Krishna Gummadi, and Virgílio Almeida. 2011. On word-of-mouth based discovery of the web. In *Proceedings of ACM IMC*.

Amanda Sorribes, Beatriz G. Armendariz, Diego Lopez-Pigozzi, Cristina Murga, and Gonzalo G. de Polavieja. 2011. The Origin of Behavioral Bursts in Decision-Making Circuitry. *PLoS Computational Biology* 7, 6 (2011).

Gabor Szabo and Bernardo A. Huberman. 2010. Predicting the Popularity of Online Content. *Commun. ACM* 53, 8 (2010), 80–88.

Dashun Wang, Zhen Wen, Hanghang Tong, Ching-Yung Lin, Chaoming Song, and Albert-László Barabási. 2011. Information Spreading in Context. In *Proceedings of WWW*.

D. J. Watts and P. S. Dodds. 2007. Influentials, Networks, and Public Opinion Formation. *Journal of Consumer Research* 34 (2007), 441–458.

Xiao Wei, Jiang Yang, Lada A. Adamic, Ricardo Matsumura de Araújo, and Manu Rekhi. 2010. Diffusion Dynamics of Games on Online Social Networks. In *Proceedings of WOSN*.

Jaewon Yang and Jure Leskovec. 2010. Modeling Information Diffusion in Implicit Networks. In *Proceedings of IEEE ICDM*.