A Simple Generative Model of Collective Online Behaviour

James P. Gleeson¹, Davide Cellai², Jukka-Pekka Onnela³, Mason A. Porter⁴, and Felix Reed-Tsochas⁵

¹, ²MACSI, Department of Mathematics & Statistics, University of Limerick, Ireland
³Department of Biostatistics, Harvard School of Public Health, Boston, USA
⁴Oxford Centre for Industrial and Applied Mathematics, Mathematical Institute, University of Oxford, Oxford OX1 3LB, UK and CABDyN Complexity Centre, University of Oxford, Oxford, OX1 1HP, UK
⁵CABDyN Complexity Centre, University of Oxford, Oxford, OX1 1HP, UK and James Martin Institute, Saïd Business School, University of Oxford, Oxford OX1 1HP, UK

June 3, 2013

Human activities—from voter mobilization [1] to political protests [2]—increasingly take place in online environments, providing novel opportunities for relating individual behaviours to population-level outcomes. The recent availability of data sets that capture the behaviour of individuals participating in online social systems has driven the emerging field of computational social science [3], as large-scale empirical data sets enable the development of detailed computational models of individual and collective behaviour. Given the inherent limitations of observational data, it is crucial to investigate the extent to which models of collective dynamics can distinguish between different individual-level mechanisms. Here we introduce a simple generative model for the collective behaviour of millions of social networking site users who are deciding between different software applications [4]. Our model incorporates two distinct components: one is associated with recent decisions of users, and the other reflects the cumulative popularity of each application. Importantly, although various combinations of the two mechanisms yield long-time behaviour that is consistent with data, only models that strongly emphasize recent popularity of applications over their cumulative popularity reproduce the observed temporal dynamics. Our approach demonstrates the value of even very simple generative models in understanding collective social behaviour, and it highlights the need to address temporal dynamics—not just long-time behaviour—when modelling complex social systems.

Choices of which movies to watch, which mobile applications (“apps”) to download, or which messages to retweet are influenced by the opinions of our friends, neighbours, and colleagues [5]. However, it is extremely challenging to distinguish between three potential explanations of observed behaviour [6]: social influence (person A adopts a behaviour, leading person B to adopt...
the same behaviour), *homophily* (*A* and *B* are friends and have similar traits, leading them to adopt a behaviour), and *confounding* (*A* and *B* share a common environment, leading them to adopt a behaviour). To circumvent such issues, it is common to examine population-level models and attempt to reproduce empirically-observed popularity distributions using the simplest possible assumptions about individual behaviour. Such efforts have arisen in a wide range of disciplines—including economics [7,8], evolutionary biology [9,10], and physics [11]—and they have recently received increased attention due to the availability of extensive data from online social networks [4,12,14].

One well-studied rule for choosing between multiple options is cumulative advantage (a.k.a. preferential attachment), in which popular options are more likely to be selected than unpopular ones. This leads to a “rich-get-richer” agglomeration of popularity [7,9,15,17]. Bentley et al. [5,18,19] proposed an alternative model, in which members of a population randomly copy the choices made by other members in the recent past. As a result, products whose popularity levels have recently grown the fastest are the most likely to be selected (whether or not they are the most popular overall). In this paper, we show that models of app-installation decisions that are biased heavily towards recent popularity rather than cumulative popularity provide the best fit to empirical data from Facebook. We use the model to identify the timescales over which the influence of Facebook users upon each others’ choices is strongest, and we argue that the interaction between these timescales and the diurnal variation in Facebook activity yields many of the observed features of the popularity distribution of apps. More generally, we illustrate how to incorporate temporal dynamics in modelling and data analysis to differentiate between competing models that produce the same long-time behaviour.

We use the Facebook apps data set that was first reported in Ref. [4]. This data includes the records, for every hour from 25 June 2007 to 14 August 2007, of the number of times every Facebook app (of the *N* = 2705 total available during this period) was installed. At the time, Facebook users had two streams of information about apps: a *cumulative-information* stream gave an “all-time best-seller” list, in which all apps were ranked by their cumulative popularity (i.e., the total number of installations to date), and a *recent-information* stream consisted of updates provided by Facebook on the recent app installation activity by the user’s friends. Users could also visit the profiles of their friends to see which applications a friend had installed.

The data thus consists of *N* time series *n*(*i*)(*t*), where the popularity *n*(*i*)(*t*) of app *i* at time *t* is the total number of users who have installed app *i* by hour *t* of the study period. The discrete time index *t* counts hours from the start of the study period (*t* = 0) to the end (*t* = 1209). The distribution of *n*(*i*) values is heavy-tailed (see Fig. S1 of the Supplementary Information (SI)), so the popularities *n*(*i*)(*t*) of the apps cover a very wide range of scales. Facebook apps first became available on 24 May 2007, corresponding to *t* ≈ −720 in our notation. By time *t* = 0, when the data collection began, 980 apps had already launched; the remaining apps in our data set were launched during the study period. Among the latter, we pay particular attention to those for which we have at least 650 hours (i.e., more than half the data collection window) of data; we call these the *Launched-Early-in-Study* (LES) apps. Denoting by *t*(*i*) the launch time of app *i*, the 921 LES apps *i* are those that satisfy *t*(*i*) > 0 and *t*(*i*) < 1209 − 650 = 559.

To measure the change in app popularity during hour *t*, we define the *increment* in popularity of app *i* at time *t* as *f*(*i*)(*t*) = *n*(*i*)(*t* + 1) − *n*(*i*)(*t*) (with *f*(*i*)(*t*) = 0 for *t* < *t*(*i*)) [4].
installation activity of users during hour $t$ is then

$$F(t) = \sum_{i=1}^{N} f_i(t) .$$  \hfill (1)

We show in the SI that $F(t)$ has large diurnal fluctuations superimposed on an aggregate growth from a mean of approximately $5.5 \times 10^4$ installations per hour at $t = 0$ to approximately twice that by $t = 1209$.

Concentrating on LES apps, we define the age-shifted popularity $\tilde{n}_i(a) = n_i(t_i + a)$ and age-shifted increment $\tilde{f}_i(a) = f_i(t_i + a)$ of app $i$ at age $a$. The age-shifted characteristics of apps enable comparison of apps when they are the same age (i.e., at the same number of hours after their launch). An examination of the trajectories of the ten largest LES apps (ordered by $\tilde{n}_i(650)$) reveals that their popularity grows exponentially for some time before reaching a steady-growth regime in which $\tilde{n}_i(a)$ increases approximately linearly with age. The corresponding age-shifted increment functions $\tilde{f}_i(a)$ reach a “plateau” at large $a$, though they have a superimposed 24-hour oscillation (see Figs. S3 and S4 in the SI).

To extrapolate from these observations to the entire set of LES apps, we scale the increment $\tilde{f}_i$ of app $i$ by its temporal average $\bar{\mu}_i = \left( \sum_{a=1}^{650} \tilde{f}_i(a) \right) / 650$ over the first 650 observations for each app. This weights very popular apps and less popular apps in a similar manner [20]. For a given set $\mathcal{I}$ of LES apps, we define the mean scaled age-shifted growth rate

$$r(a) = \left\langle \frac{\tilde{f}_i(a)}{\bar{\mu}_i} \right\rangle_{\mathcal{I}},$$  \hfill (2)

where $\langle \cdot \rangle$ denotes an ensemble average over all the apps in the set $\mathcal{I}$.

Mean scaled age-shifted growth rates reveal several interesting features (see Fig. 1a). First, at large ages (e.g., $a \geq 150$ hours), the function $r(a)$ has 24-hour oscillations superimposed on a nearly constant curve. The behaviour of $r(a)$ is very different for smaller ages ($a < 150$ hours); we dub this the novelty regime, as it represents the (approximately one-week) time period immediately following the launch of apps. The $r(a)$ curve for the entire LES set is similar to those found by splitting the LES set into two disjoint subsets—the 460 applications with earlier launch times ($t_i \leq 260$; early-launch) and the 461 applications with later launch times ($t_i \geq 261$; late-launch). The small difference between the $r(a)$ curves for these cases gives an estimate of the inherent diversity within the data and sets a natural target for how well stochastic simulations can be fit to the data (see SI3).

To directly measure the growth of new apps in their first 650 hours, we show the distribution of $\tilde{n}_i(650) - \tilde{n}_i(0)$ for the entire LES set in Fig. 1b. We also show the corresponding distributions for the two LES subsets. The similarity of distributions for early-born apps and late-born apps implies that age, at least in the range examined here, does not have a strong effect on the growth of new apps. This contrasts with Yule-Simon models of popularity [7,16,21] and related preferential attachment models used to model citations [11].

The popularity dynamics for the novelty regime ($a < 150$ hours) seems to be app-specific (see Figs. 1b and S4), but a simple model can satisfactorily describe the post-novelty regime. We introduce a general stochastic simulation framework with a history-window parameter $H$ and
Figure 1: Mean scaled age-shifted growth rate $r(a)$ (left column) and complementary cumulative distribution function (CCDF) for the number of app installations (right column). (a, b) Behaviour of the entire LES set of applications and its two subsets (which are described in the text). (c,d) Cumulative-information model ($\gamma = 1$), for which we show CCDFs for popularity at $t = 1209$ (upper symbols) and for LES app growth to age $a = 650$ (lower symbols). (e,f) Recent-information model with short memory ($\gamma = 0$, $H = 168$, $T = 5$). (g,h) Recent-information model with long-memory ($\gamma = 0$, $H = 168$, $T = 50$). In panels c–g, we show the empirical data in black.
consider an app to be within its *history window* for the first $H$ hours that data on the app is available. The history window of LES apps extends from their launch time to $H$ hours later; for non-LES apps, we define the history window to be the first $H$ hours ($t = 0$ to $t = H$) of the study.

We conduct stochastic simulations by modelling $F(t)$ computational “agents” in time step $t$, each of whom installs one app at that time step. Note that our simulated agents do not correspond directly to Facebook users, as we do not have data at the level of individual users. In reality, a Facebook user may, for example, install several different apps during an hour; in our simulations their actions would be modelled by the choices of several computational agents.

We simulate the choices of the agents as follows. First, for any app $i$ that is in its history window at time $t$, the increment $f_i(t)$ is copied directly from the data. This determines the choices of $F_H(t)$ of the agents, where $F_H(t)$ is the number of installations of all apps that are within their history window at time $t$. Each of the remaining $F(t) - F_H(t)$ agents then installs any one of the apps that are not in their history window. An installation probability $p_i(t)$ is allocated based on model-specific rules (see below), and the $F(t) - F_H(t)$ agents each (independently) choose app $i$ with probability $p_i(t)$. These rules ensure that the total number of installations in each hour exactly matches the data and that the history window of each app is reproduced exactly.

We investigate several possible choices for $p_i(t)$ by comparing the results of simulations with the characteristics of the data highlighted in Figs. 1a,b. The history window parameter $H$ plays an important role in capturing the app-specific novelty regime. However, if $H$ is very large, then most of the simulation is copied directly from the data and the decision probability $p_i(t)$ becomes irrelevant. It is therefore desirable to find models that fit the data well while keeping $H$ as small as possible. Motivated by the information streams available to Facebook users during the data collection period, we propose a model based on a combination of a *cumulative rule* $p^c_i(t)$ and a *recent rule* $p^r_i(t)$.

Agents who use the *cumulative rule* choose app $i$ with a probability proportional to its cumulative popularity $n_i(t - 1)$, yielding

$$p^c_i(t) = K n_i(t - 1) ,$$

where the constant $K$ is determined by the normalization $\sum_i p^c_i(t) = 1$. In contrast, an agent who follows the *recent rule* copies the installation choice of an agent who acted in an earlier time step, with some memory weighting (see Eq. (4) below). Thus, apps that were recently installed by many agents (i.e., apps with large $f_i(\tau)$ values for $\tau \approx t$) are more likely to be installed at time step $t$ even if these apps are not yet globally popular (i.e., $n_i(t - 1)$ can be small). In reality, the information available to Facebook users on the recent popularity of apps was limited to observations of the installation activity of their network neighbours. As we lack any information on the real network topology, we make the simplest, mean-field, assumption: that the network is sufficiently well-connected to enable all agents in the model to have information on the aggregate (system-wide) installation activity. Agents applying the recent rule then choose app $i$ with a probability proportional to the recent level of that app’s installation activity:

$$p^r_i(t) = L \sum_{\tau=0}^{t-1} W(t - \tau) f_i(\tau) ,$$

where $L$ is determined by the normalization $\sum_i p^r_i(t) = 1$. The *memory weighting function* $W(\tau)$...
Figure 2: Parameter planes showing the $L^2$ norm of the difference between the simulated $r(a)$ curve and the $r(a)$ curve from the data for the recent-information-dominated model described in the text. The parameter $H$ is the length of the history window, and $T$ is the characteristic time of the exponential response-time distribution. For each point in the plane, we average values of the $L^2$ norm over 24 realizations; all values above 3.11 are shown as dark red.

determines the weight assigned to activity from $\tau$ hours ago and thereby incorporates human-activity timescales [22]. In the SI, we consider several examples of plausible memory functions and also examine the possibility of heterogeneous app fitnesses.

Note that if our data set included the early growth of every app, than a constant weighting function $W(t) \equiv 1$ would reduce $p^c_i$ to $p^r_i$. However, our finite data window means that many apps have large values of $n_i(0)$, and so we cannot capture the cumulative rule by using a suitable weighting function in the recent rule. Instead, we introduce a tunable parameter $\gamma$, with $0 \leq \gamma \leq 1$, so that the population-level installation probability $p_i$ used in the simulation is a weighted sum of the cumulative rule and the recent rule:

$$p_i(t) = \gamma p^c_i(t) + (1 - \gamma) p^r_i(t).$$

To explore the model, we start by considering the case $\gamma = 1$, in which agents consider only cumulative information. In Figs. 1c,d, we compare the results of stochastic simulations with the data (see Figs. 1a,b) using a history window of $H = 168$ hours (i.e., 1 week). Clearly, the cumulative-information model does not match the data well. Although the app popularity distributions at $t = 1209$ are reasonably similar (see Fig. 1d), the largest popularities are overpredicted by the model. By contrast, the popularity of the LES apps—which include many of the less popular apps—is underpredicted. In particular, their mean scaled age-shifted growth rate has a lower long-term mean than that of the data. In the SI, we show that an alternative cumulative-information model based on the ranking of apps by their popularity [23] also matches the data poorly.

We next consider the case in which $\gamma$ is small, so that recent information dominates [5,19]. In Fig. 2 we show results for stochastic simulations using an exponential response-time distribution $P(t) = \frac{1}{T} e^{-t/T}$ (see SI4) to determine the weights $W(t)$ assigned to activity from $t$ hours earlier, for varying history-window lengths $H$ and response-time parameters $T$. The colours in the $(H,T)$ parameter plane represent the $L^2$ norm of the difference between the simulated $r(a)$ curve and the $r(a)$ curve from the data. A value of 3.11 is representative of inherent fluctuations in the data (see SI3), and the bright colours in Fig. 2 represent parameter values for which the difference
between the model’s mean growth rate and the empirically observed growth rate is less than the magnitude of fluctuations present in the data. Observe that the model requires a history window of approximately 1 week (i.e., $H \approx 168$ hours) to match the data. As $\gamma$ increases, cumulative information is weighted more heavily, and the region of good-fit parameters moves towards larger $T$ and larger $H$ (see also SI3). As noted previously, large-$H$ models trivially provide good fits (because they mostly copy directly from data), but the $\gamma = 0$ case provides a good fit to the data with a relatively short history window $H$.

In Fig. 1e,f, we compare model results with data for parameter values $H = 168$, $T = 5$, and $\gamma = 0$ (i.e., the “recent-information, short-memory” case). This reproduces the app popularity distributions of the data rather well, but the mean scaled age-shifted growth rates are markedly different. In contrast, Figs. 1g,h compares model results with data for parameter values $H = 168$, $T = 50$, and $\gamma = 0$ (i.e., the “recent-information, long-memory” case). These parameters are just inside the “good-fit” region of Fig. 2a, so the $r(a)$ curve in Fig. 1g matches the data well. Moreover, the popularity distributions at $t = 1209$ and at age 650 (see Fig. 1h) are both reasonably matched by the model. These considerations highlight the importance of using temporal data to develop and fit models of complex systems. Distributions at single times can be insensitive to model differences, and the $r(a)$ curves are crucial for distinguishing between competing models. In the SI, we provide additional comparisons of the models by considering turnover within the top-10 apps. We also show that the recent-information ($\gamma = 0$) case still gives good fits to the data if the exponential response-time distribution is replaced by a lognormal, gamma, or uniform distribution.

Another noteworthy feature of the recent-information model is its ability to produce heavy-tailed popularity distributions in stochastic simulations even if no history is copied from the data ($H = 0$). Even if all apps initially have the same number of installations, random fluctuations lead to some apps becoming more popular than others, and the aggregate popularity distribution becomes heavy-tailed [10,18,19,24]. In the SI, we show that this situation is described by a near-critical branching process, for which the mean-field asymptotic popularity distribution is the power law $P(n) \sim n^{-\alpha}$, where $\alpha \in [3/2, 2)$ [25–29].

Our model suggests that app adoption among Facebook users was guided more by recent popularity of apps (as reflected in installations by friends within 2 days) than by cumulative popularity. The fact that the model is a near-critical branching process might help explain the prevalence of heavy-tailed popularity distributions (with power-law exponents $\alpha < 2$) observed in information cascades on social networks, such as the spreading of retweets on Twitter [2,13,14] or news stories on Digg [30]. Our simulation-based approach also highlights the need to address temporal dynamics when modelling complex social systems. A complementary approach to simulations are experiments, in which the randomized assignment of subjects to treatment and control conditions removes many of the difficulties of making causal inference using observational data. The obvious downside of offline experiments is their cost. Online experiments have been used successfully in computational social science [3], but it is challenging to run experiments in online environments that people actually use (as opposed to creating new online environments with potentially distinct behaviours). If longitudinal data is available, as in the present case, a model’s fit can be evaluated based not only on long-time behaviour but also on dynamical behaviour. Given that several models successfully produce similar long-time behaviour, the investigation of temporal dynamics is critical for distinguishing between competing models. As more observational
data from online social networks becomes available, we believe that this modelling strategy, which leverages temporal dynamics, will become increasingly essential.

Acknowledgements

We thank Andrea Baronchelli, Ken Duffy, James Fennell, James Fowler, Sandra González-Bailón, Stephen Kinsella, Yamir Moreno, Peter Mucha, Puck Rombach, and Frank Schweitzer for helpful discussions, and the SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities. Funding is acknowledged from Science Foundation Ireland (JPG, DC), the FET-Proactive project PLEXMATH FP7-ICT-2011-8 grant #317614 (JPG, DC, MAP), and the John Fell Fund from University of Oxford (MAP).
SUPPLEMENTARY INFORMATION

SI1: Data Cleaning and Aggregate Installation Activity

The data was downloaded from Facebook for all existing 2720 applications (“apps”) between 25 June 2007 (shortly after applications were introduced) and 14 August 2007 [4]. The data consists of time series \( n_i(t) \), where \( i \in \{1, 2, \ldots, 2720\} \), discrete time is indexed by the (real-time) hour \( t = 1, 2, \ldots, 1209 \), and \( n_i(t) \) corresponds to the aggregate number of users who have application \( i \) installed at time \( t \). Data for 15 applications was corrupted, so we omitted these from our investigation and examined a total of 2705 applications. This data covers 100% of the population of 50 million potential adopters and about 99% (2705 of 2720) of all applications that can be adopted. This thereby gives (almost) a complete view of system-wide adoptions during the time period of the data collection. We define the launch time \( t_i \) of app \( i \) as the smallest value of \( t \) for which \( n_i(t) > 0 \), and we define the increment in hour \( t \) for app \( i \) to be \( f_i(t) = n_i(t+1) - n_i(t) \).

The data-cleaning process involves removing any undefined values within the data and imputing replacement values. For each app \( i \), if \( f_i(t) \) is undefined for \( t > t_i \), then we copy the most recent well-defined increment value for app \( i \) into \( f_i(t) \). A second cleaning step entails removing negative values of \( f_i(t) \). Such values correspond to the (rare) cases in which deinstallations exceeded installations of an app in a given hour. We do this by setting any instances in the data with \( f_i(t) < 0 \) to \( f_i(t) = 0 \). The effects of the data cleaning are small in the context of the aggregate statistical characteristics of the data. In Fig. S1 the complementary continuous distribution function (CCDF) at \( t = 1209 \) for the cleaned data is shown in black, while the corresponding case using the raw (pre-cleaning) \( n_i(t) \) time series is shown as red circles. The two distributions are almost indistinguishable, except for the smallest (i.e., least popular) apps, indicating that the
cleaned data is very similar to the original data.

Figure S1 shows that the popularities $n_i(t)$ of the apps cover a range of scales from very small to extremely popular and that the distribution of $n_i$ values is heavy-tailed. In Fig. S2, we show the total app installation activity $F(t)$, which is defined by Eq. (1) of the main text, of Facebook users during hour $t$. This function exhibits slow growth and 24-hour oscillations. We highlight these features by also plotting a linear growth function $A(t) = c_1 + c_2 t$ and (as a guide to the eye) a growing oscillation $A(t)\psi(t)$, where $\psi(t) = 1 + 0.5 \cos(2\pi(t + 8)/24)$ gives the oscillatory part of the function. Least-squares fitting gives $c_1 = 5.5 \times 10^4$ and $c_2 = 49$, meaning that by $t = 1209$—the end of the data collection period—the average hourly installation rate is approximately twice as large as it was at $t = 0$.

SI2: Top Ten Launched-Early-in-Study (LES) Apps

In Fig. S3, we show the ten most popular Launched-Early-in-Study (LES) apps. We order them by $\tilde{n}_i(650)$, which denotes the number of installations by age 650. To highlight common features of app growth, we use the heuristic fitting function

$$m(a) = \begin{cases} 
C(e^{Da} - 1), & \text{if } a \leq \theta, \\
C(e^{D\theta} - 1) + E(a - \theta), & \text{if } a > \theta,
\end{cases} \quad (S1)$$

where the parameters $C$, $D$, $E$, and $\theta$ are determined by least-squares fitting of $m(a)$ to $\tilde{n}_i(a) - \tilde{n}_i(0)$ in each case. We give the values of these parameters in Table S1 The parameter values that we obtain are sensitive to the initial guesses used in the fitting routine, but it is nevertheless clear that most apps exhibit exponential growth in a novelty regime (i.e., when age $a < \theta$) followed by linear growth at later ages (i.e., $a > \theta$).

In Fig. S4, we show the scaled age-shifted growth rates $\tilde{f}_i(a)/\tilde{\mu}_i$ for the three most popular LES apps and the mean scaled age-shifted growth rate $r(a)$ for the set of top-20 LES apps. At large values of $a$, the function $r(a)$ is qualitatively similar to that of the full LES set in Fig. 1a in the main text, as it exhibits a “quasi-stationary” (i.e., constant plus 24-hour oscillations) behavior.

The notable exception among the top 10 in terms of fitting quality is Harry Potter Magic Spells (the 6th most popular app).
Figure S3: Growth trajectories of the 10 most popular LES apps, ordered by their popularity when at an age of 650 hours. The data values are in black, and red dashed curves show the fitting function $m(a) + \tilde{n}_i(0)$ described in Eq. (S1) with parameter values from Table S1.
Figure S4: Scaled age-shifted growth rate functions $\tilde{f}_i(a)/\tilde{\mu}_i$ for the three most popular LES apps and the mean scaled growth rate over the 20 most popular LES apps.

| Rank | Name                             | $C$            | $D$            | $E$            | $\theta$ |
|------|----------------------------------|----------------|----------------|----------------|----------|
| 1    | Likeness                         | $1.59 \times 10^5$ | $1.48 \times 10^{-2}$ | $5.63 \times 10^3$ | 35       |
| 2    | FunWall                          | $4.25 \times 10^4$ | $5.26 \times 10^{-3}$ | $3.31 \times 10^4$ | 481      |
| 3    | What’s your stripper name?       | $4.10 \times 10^3$ | $1.12 \times 10^{-2}$ | $2.30 \times 10^3$ | 243      |
| 4    | My Aquarium                      | $4.44 \times 10^3$ | $2.85 \times 10^{-2}$ | $2.10 \times 10^3$ | 99       |
| 5    | Vampires                         | $3.14 \times 10^4$ | $2.58 \times 10^{-2}$ | $1.77 \times 10^4$ | 48       |
| 6    | Harry Potter Magic Spells        | $3.10 \times 10^4$ | $8.30 \times 10^{-3}$ | $1.22 \times 10^3$ | 350      |
| 7    | Pirates vs. Ninjas               | $7.65 \times 10^4$ | $1.23 \times 10^{-2}$ | $1.70 \times 10^4$ | 236      |
| 8    | Booze Mail                       | $7.22 \times 10^4$ | $9.95 \times 10^{-3}$ | $1.88 \times 10^4$ | 329      |
| 9    | Superlatives                     | $1.14 \times 10^4$ | $7.24 \times 10^{-3}$ | $1.50 \times 10^4$ | 402      |
| 10   | Texas HoldEm Poker               | $1.10 \times 10^4$ | $8.24 \times 10^{-3}$ | $1.02 \times 10^4$ | 399      |

Table S1: Parameter values for the fitting functions $m(a)$ used in Fig. S3.
behaviour. However, the small-α novelty regime is different in the two cases, and this reflects differences in early-stage growth patterns. In particular, the most popular apps show steadily growing popularity during the novelty regime. This is consistent with the exponential growth in Fig. S3, but it contrasts with the decrease in novelty experienced by the majority of apps in their early stages and reflected in the r(α) curve of Fig. 1a of the main text.

SI3: Discussion of $L^2$ Error in the Mean Scaled Age-Shifted Growth Rate

In Fig. 1a of the main text, the mean scaled age-shifted growth rate $r(α)$ for the entire LES set is seen to be similar to the corresponding $r(α)$ curves found by splitting the LES set into two disjoint subsets: the early-launch subset, and the late-launch subset. To quantify the level of inherent diversity within the data, we calculate the $L^2$ norm of the difference between the $r(α)$ curves, and call this the $L^2$ error of the subset:

$$E_{L^2} = \sqrt{\sum_{α=1}^{650} (r_{LES}(α) - r_{subset}(α))^2}. \quad (S2)$$

We find that for the subsets described above the $L^2$ error is less than 3.11, and we take this figure to represent a natural target for how well stochastic simulations can be fit to the data. In Fig. 2 of the main text, we therefore show all $L^2$ error values above 3.11 as dark red, and concentrate on the light-coloured regions of the (H, T) parameter plane where high-quality fits are possible. In Fig. S5 we show the $L^2$ error $E_{L^2}$ as a function of the memory time $T$ for exponential memory weighting function $W(τ)$, for fixed history window length $H = 168$, and several values of the parameter $γ$, cf. Fig. 2 of the main text. The dashed line indicates the threshold for the “good-fit” regime of $E_{L^2} \leq 3.11$. The error tends to increase with increasing $γ$ and is unacceptably high for all values of $T$ for $γ > 0.2$.

SI4: Response Functions Generating Memory Weighting

If one assumes that the total installation activity $F(t)$ is constant in time, then the memory weighting function $W(τ)$ introduced in Eq. (4) of the main text is proportional to the probability that an agent copies the installation choice of an agent from $τ$ hours in the past (see SI5 for details). Consequently, we consider weighting functions that are related to previous empirical studies of the distribution of response times for e-mails [31–33]. Consider an update message that informs a Facebook user—here modelled as a single computational agent—that a friend has installed a certain app, which can then lead to the user subsequently installing the app. Let $τ'$ denote the time between receiving the update message and installing the app, and let $P(τ')$ denote the probability distribution function (PDF) of these “response times” across the user population. We coarse-grain to the one-hour time resolution of the data by setting $W(τ) = \int_{τ-1}^{τ} P(τ') dτ'$ for $τ = 1, 2, \ldots$, with $W(0) = 0$.

In the main text, we showed an example in which $P(t)$ is an exponential distribution. We now consider alternative assumptions on the underlying response-time distribution $P(t)$ and show
Figure S5: The $L^2$ error in the mean scaled age-shifted growth rate $r(a)$ as a function of the memory time $T$ for exponential memory weighting function $W(\tau)$, history window $H = 168$, and several values of the parameter $\gamma$. Each point is the result of a single realization of the stochastic simulation model.

We find similar results for lognormal, gamma, and uniform distributions. In all of these cases, we obtain good results with a history window parameter of $H \approx 168$ hours (i.e., 1 week). Interestingly, when $H = 168$, the results for all distributions are very similar to those shown in Fig. 1g,h of the main text if the characteristic response-time $\langle \tau \rangle = \sum_{\tau=1}^{168} \tau W(\tau) / \sum_{\tau=1}^{168} W(\tau)$ is about 45 hours (i.e., approximately 2 days).

In Fig. S6, we show results for the uniform distribution given by

$$P(t) = \begin{cases} \frac{1}{T}, & \text{if } t \leq T, \\ 0, & \text{if } t > T, \end{cases}$$

(S3)

where $T$ is the cutoff time. (The mean response time is $T/2$.) As with Fig. 2 in the main text, we show results in the $(H,T)$ parameter plane to highlight the roles of both the history window $H$ and the memory cutoff $T$. The three panels illustrate the effects of using increasing amounts of cumulative information (i.e., progressively larger values of $\gamma$) in the installation probability $p_i$. Moving from left to right, the weighting of cumulative-information increases from 0 to 0.1 and 0.15. As this weight increases, observe that the “good-fit” region of parameters moves to higher $H$ values and higher $T$ values. This supports our conclusion in the main text that the

$^2$The value of $\langle \tau \rangle$ is similar to the mean response time if most of the probability mass lies in the range $\tau < H$. The cutoff at $\tau = H$ reflects the fact that apps at early stages in their simulated growth possess a window of only approximately $H$ hours of previous-installation history to drive their evolution.
recent-information case $\gamma = 0$ is “optimal” in the sense of requiring only a relatively small history window size $H$ to fit the data. Similar conclusions were also reached in Ref. [34]. We have also confirmed that the other main results for the exponential distribution (e.g., the ones depicted in Fig. 1 in the main text) are closely reproduced using the uniform distribution (where we set $T \approx 100$ so that the mean response times are equal in the two cases).

In Fig. S7, we show results in which the response-time distribution $P(t)$ is given by lognormal (top row) and gamma (bottom row) distributions. These distributions both have two parameters, so we fix the history window size $H$ at 168 hours (i.e., 1 week) and consider the effect of the parameters that define the distributions. The lognormal distribution with parameters $\mu$ and $\sigma$ is

$$P(t) = \frac{1}{t \sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(\ln t - \mu)^2}{2\sigma^2} \right\},$$

and it was used in Refs. [31, 32] to fit distributions of e-mail response times. The gamma distribution [32, 33] is

$$P(t) = \frac{1}{\Gamma(k)T^k} t^{k-1} e^{-\frac{t}{T}},$$

which for $k < 1$ becomes a power-law distribution as $T \to \infty$ and for $k = 1$ is an exponential distribution. The middle and right panel of each row of Fig. S7 gives results for the cases $\gamma = 0$ and $\gamma = 0.1$, respectively. For $\gamma = 0.15$, the “good-fit” regions have almost disappeared from these plots, so we do not show them. The left panel of each row shows the contours of the quantity

$$\langle \tau \rangle = \frac{\sum_{\tau=1}^{168} \tau W(\tau)}{\sum_{\tau=1}^{168} W(\tau)},$$

which is related to the goodness-of-fit of the recent-information ($\gamma = 0$) models. Observe that the light-coloured regions of the middle panels align closely with the contours showing $\langle \tau \rangle$ values between 30 and 50 hours. Note that $\langle \tau \rangle$ is not identical to the mean response time of the distribution $P(t)$ because of the cutoff at 168 hours in the sums of Eq. (S4). Without these cutoffs, the mean of (for example) the lognormal distribution grows exponentially as $\mu$ and $\sigma$ increase. We believe that this cutoff reflects the fact that the history window of 168 hours defines the $\tau$ range upon which the recent-information model operates for an app that was launched.
Figure S7: As Fig. 2 in main text but for lognormal distribution (top row) and gamma distribution (bottom row) of response times. The first panel in each row shows the contours of the cutoff mean response time $\langle \tau \rangle$ defined in Eq. (S4).

recently. It seems that a memory weighting corresponding to roughly 2 days (i.e., 48 hours) of recent activity is sufficient in all of these cases to fit the recent-information model to the data.

SI5: Recent-Information Model as a Random-Copying-with-Memory Process

In this section, we show that the recent-information model ($\gamma = 0$) described in the main text can be interpreted as a random-copying process that is similar to those studied by Bentley et al. [18,19]. We also describe these models in terms of branching processes and discuss the circumstances under which one obtains critical branching processes, in which each parent has, on average, one child over its lifetime [25].

We consider a random-copying model in which each individual (an agent in our simulation) at time $t$ copies the action (i.e., the choice of app to install) of an agent from a previous time step. In the schematic of Fig. S8a, we denote the copying action with an arrow from the earlier installation event to the later installation event (i.e., arrows point from the target of the copying to the copier). This generates a tree structure in time in which each node represents a single installation action and each arrow links a “parent” (target) node to some number of “child” (copier) nodes. Each child node has exactly one parent—this represents the installation action that was copied—but the number of children assigned to any given parent depends on the details of the random-copying process. As noted in the main text, we do not have any information on
the network topology, so we make the mean-field assumption that all agents can copy the action of any earlier agent, unrestricted by network connectivity.

There are $F(t)$ agents who install an app at time $t$, and all act independently of each other. Consider one such agent $Y$ who must choose an earlier installation to copy. Let $\Phi(X,Y)$ denote the probability that $Y$ copies the past action of a selected node $X$ (see Fig. S8a). Normalization implies that $\sum_X \Phi(X,Y) = 1$, where the sum is over all possible target nodes $X$ such that $X$ takes an action before $Y$. We assume that the selection probability depends only on the time $\tau$ of the target node $X$ and the time $t$ of the action $Y$, so we write $\Phi(X,Y) = \phi(\tau,t)$. This implies that all installations at time $\tau$ are equally likely to be copied by $Y$. Moreover, we assume that the dependence on $\tau$ appears only through the time $t_e := t - \tau$ elapsed since the target event, so $\phi(\tau,t) \propto W(t - \tau)$, where $W$ is the memory weighting function (see SI4). Because there are $F(\tau)$ installing agents (i.e., nodes) at time $\tau$, the correctly normalized copying probability must obey $\sum_{\tau < t} \phi(\tau,t)F(\tau) = 1$. This yields

$$\phi(\tau,t) = \frac{W(t - \tau)}{\sum_{\tau'=-\infty}^{t-1} W(t - \tau')F(\tau')}.$$  \hspace{1cm} (S5)

Note we are allowing a potentially infinite history here, which might be appropriate for very heavy-tailed memory-functions \([35, 36]\).

Using this random-copying model, we want to compute the probability that user $Y$ installs a given app $i$ at time $t$. There are $f_i(\tau)$ agents who install app $i$ at each time $\tau$ with $\tau < t$. (Installer $X$ in Fig. S8a is just one example of many.) Each of these agents can be copied by agent $Y$ with probability $\phi(\tau,t)$. Summing over all earlier times implies that the total probability that $Y$ installs app $i$ is

$$\sum_{\tau = -\infty}^{t-1} \phi(\tau,t)f_i(\tau).$$  \hspace{1cm} (S6)
Using the definition of $\phi(\tau, t)$, (S6) can be rewritten as
\[
\sum_{\tau = -\infty}^{t-1} W(t-\tau) f_i(\tau) \over \sum_{\tau = -\infty}^{t-1} W(t-\tau') F(\tau'),
\]
which is precisely $p_i^\tau$ in the main text, where we note that $f_i(\tau) = 0$ for $\tau < 0$ in Eq. (4) from the main text because data is available only from $t = 0$ onwards. The normalization constant $L$ in Eq. (4) can be written as
\[
L = \left( \sum_i \sum_{\tau'} W(t-\tau') f_i(\tau') \right)^{-1}.
\]
by reordering the summations and using Eq. (1) from the main text.

Returning to the branching-process interpretation of Fig. S8a, we calculate the expected number of children for each parent in the tree. Consider node $X$, which can be copied by any one of the $F(t)$ installing agents at time $t$. Each of these agents, comprising potentially many children of $X$ (such as $Y$), chooses to copy $X$ with probability $\phi(\tau, t)$. Summing over $t$ gives the expected number of children of node $X$ (and indeed of any user at time $\tau$) over all future times:
\[
z(\tau) = \sum_{t=\tau+1}^{\infty} \phi(\tau, t) F(t) = \sum_{t=\tau+1}^{\infty} \frac{W(t-\tau) F(t)}{\sum_{\tau' = -\infty}^{t-1} W(t-\tau') F(\tau')}.
\]
This effective branching number $z(\tau)$ depends on the time $\tau$ of the parent node, because the interaction of the variable level of installation activity $F(t)$ with the memory function $W(\tau)$ implies that installations from some times are more likely to be copied in the future than installations from other times.

Note that if $F(t)$ is constant, then $z(\tau) = 1$ for all $\tau$. (Letting $s = t - \tau$ and $s' = t - \tau'$ gives $z(t) = \sum_{s=1}^{\infty} W(s)/\sum_{s'=1}^{\infty} W(s') = 1$ in this case.) Because each individual installation then has, on average, exactly one offspring, we obtain a critical branching process [25], for which power-law distributions of popularity (with exponents $\alpha \in [3/2, 2]$) are expected in the mean-field limit [26, 27, 29]. Thus, the competition among apps for the finite number of installer slots leads to a critical branching process that is reminiscent of the self-organization mechanism in self-organized-criticality (SOC) models [26, 37]. Bentley et al. [19] used numerical computations to examine this case of constant-$F(t)$, though they did not give a branching-process interpretation. Note additionally that we concentrate on the accumulated popularity over time $n_i(t)$. In contrast, rewiring models such as those in Refs. [24, 38] focus instead on the distribution of short-time increments (similar to our $f_i(t)$).

As shown in Fig. S2, the total installation activity $F(t)$ exhibits substantial variation over time due to daily human activity patterns and to the aggregate growth in popularity of Facebook
applications. In Fig. S8b, we show the effective branching number \( z(t) \) calculated from Eq. (S8) using the \( F(t) \) function taken from the data and the long-memory weighting function (i.e., an exponential response-time distribution with \( T = 50 \) hours) used in Figs. 1g,h of the main text. Despite the growth and fluctuations in \( F(t) \) that are evident in Fig. S2, the values of \( z(t) \) remain close to the critical value of 1 throughout the period of the study. This occurs because the memory of the weighting function \( W \) achieves a balance: it is sufficiently long so that it dampens the impact of daily oscillations on \( z(t) \), but it is sufficiently short so that it also ameliorates the effect of the slow growth in \( F(t) \) on \( z(t) \). The resulting branching process is therefore near-critical [27], with effective branching number between 0.9 and 1.2, and this might help explain the heavy-tailed popularity distributions observed in this (and many other) empirical data sets [5]. Related models for directed networks like Twitter have also recently been shown to be poised at criticality [39, 40].

SI6: Ranking Model

The ranking model introduced in Refs. [23, 41] suggests an alternative rule for how Facebook users might choose an app to install if they base their decisions only on a global listing of all apps by their popularity. If an agent focuses only on the rank order of apps within the list and ignores the popularities (i.e., the numbers of installations) of the apps, then it is plausible that the probability of choosing app \( i \) at time \( t \) depends only on its ranking at time \( t-1 \). In the ranking model, this probability is

\[
p^r_i(t) = \frac{r_i^{-\delta}}{\sum_{j=1}^{t-1} r_j^{-\delta}},
\]

where \( r_i \) is the rank of app \( i \) at time \( t-1 \) and the quantity \( \delta \) is a tunable parameter. For example, the second-ranked app (\( r_i = 2 \)) is thus \( 2^\delta \) times less likely to be chosen than the top-ranked app (\( r_i = 1 \)). Such rich-get-richer dynamics is different to the linear preferential attachment mechanism of Eq. (3) of the main text, though it can also lead to power-law distributions of popularity [23, 41].

In Fig. S9, we show the results of replacing the cumulative-information rule of Eq. (3) from the main text with the ranking model rule (S9) while neglecting all recent information (i.e., putting \( \gamma = 1 \)). For \( \delta = 2 \), the ranking model results are qualitatively similar to those of the linear preferential attachment case of Fig. 1c,d of the main text. Both models underpredict installations of LES apps, so the \( r(a) \) curve is too low at large ages. For \( \delta = 1 \), however, installations of (less-popular) LES apps are overpredicted by the ranking model, so the \( r(a) \) curve in Fig. S9 is higher than the data curve at large \( a \). In all cases—even \( \delta = 1.5 \), for which the fit to \( r(a) \) is reasonably good—the distributions of app popularities differ dramatically from the data [see Fig. S9b]. We conclude that the ranking model, like the cumulative-information model considered in the main text, cannot provide a good fit to the data on Facebook apps.

SI7: Turnover in the Top 10

In Fig. S10, we examine turnover in the top-10 list of apps [19, 42] by plotting the trajectories of the largest apps over the duration of the study and comparing them with the results of various
models. Figure S10a is based on the full data set and shows the popularity $n_i(t)$ of apps that are in the top 10 (ranked by their popularity at time $t = 1209$). The solid curves (and legends) highlight the apps that were in the top 5 at $t = 0$. By comparing Fig. S10a (the ground truth) with the corresponding plots for realizations of the various models [see Figs. S10b,c,d], we obtain some insight into how well each model captures relative growth rates and turnover among the most popular apps.

In Fig. S10b, we show the app-popularity trajectories for the cumulative-information model of Fig. 1c,d of the main text. In Fig. S10c, we show them for the ranking model discussed in Section SI6. Both are cumulative-advantage models, which is evident in the fact that the ordering among the top-10 apps is unchanged over time. In contrast, as shown in Fig. S10d, the recent-information model of Fig. 1g,h from the main text allows turnover levels that are closer (though, of course, not identical in their details) to those in the empirical data.

SI8: Heterogeneous Fitnesses

We now consider replacing the recent rule [Eq. (4) of the main text] with an alternative that includes a fitness parameter $\lambda_i$ for app $i$. The refined recent rule is

$$p_i^r(t) = \lambda_i L \sum_{\tau=0}^{t-1} W(t-\tau)f_i(\tau), \quad (S10)$$

which is normalized such that $\sum p_i^r(t) = 1$. Apps with higher fitnesses are more likely (all else being equal) to be selected for installation than apps with lower fitnesses. Thus far, we have focussed on the so-called neutral-model [5] case, in which all fitnesses are equal (with $\lambda_i = 1$ for all $i$). Noting from Fig. 1h of the main text that some of the largest LES app popularities are underpredicted by the otherwise successful recent-information, long-memory model with homogeneous fitnesses (e.g., for $n \in [10^5, 10^6]$), it is natural to ask whether heterogeneous fitnesses might lead to a better fit to the data.
Figure S10: Growth trajectories of the 10 most popular applications in (a) data, (b) simulation with cumulative-information model $\gamma = 1$, (c) simulation with ranking model (with $\delta = 1$), and (d) simulation with recent-information, long-memory model described in the main text. The legends give the rank at $t = 0$ in parentheses followed by the rank at $t = 1209$ for the top-5 applications from $t = 0$ (solid curves). The dashed curves show the popularities of the remainder of the top-10 apps from $t = 1209$. In all cases, the history window is $H = 168$ hours (i.e., 1 week).
In Fig. S11, we show the growth of ‘Pirates vs. Ninjas’, the 7th most popular (at age $t = 650$) LES app (see panel 7 of Fig. S3). This is one of the apps in which the recent-information, long-memory model of the main text with a 1-week history window gives an inaccurate prediction (solid red curve). This leads to notable differences between the CCDF of LES apps in Fig. 1h near $n \approx 10^6$. We thus consider changing the fitness of this particular app to a value $\lambda_P$ exceeding 1, while maintaining the $\lambda_i$ values for all other apps at unity. In Fig. S11a, we show the results of typical simulations using the dashed red curves. Although it is clearly possible to increase the popularity of this app by changing its fitness, we note that the $\lambda_P > 1$ trajectories exhibit increasing curvature, and the growth tends towards exponential in time rather than linear in time. For comparison, we also show results of an equal-fitness simulation in which we use a larger history window of 2 weeks (i.e., $H = 336$ hours) for all apps. In this case, the model’s linear growth is much closer to the data, because the history window now includes the transition from novelty to post-novelty regimes (see Table S1 and the heuristic fit of Fig. S3) at about 236 hours (i.e., 1.4 weeks). Figure S11b confirms that using this longer history window leads to a much closer match between model and data.

We conclude that there does not appear to be strong evidence for heterogeneous fitnesses [as defined in our model through Eq. (S10)] among the apps, at least in the post-novelty regime. This conclusion is consistent with the findings of Bentley et al. regarding the applicability of the neutral model to other instances of choice among multiple alternatives [18] as well as with the experimental result of Salganik et al. [43], who showed that attractiveness of downloaded music is influenced more heavily by the actions of other downloaders than by the inherent quality of the music itself.
In Fig. S12a, we show a fluctuation-scaling (FS) plot of the Facebook apps data. As in Ref. [4], we calculate for each app $i$ the mean $\mu_i$ and standard deviation $\sigma_i$ of the increments $f_i(t)$ over times $t$ from launch time $t_i$ to the end of the data ($t = 1209$). We then plot $\mu_i$ versus $\sigma_i$ for all $i$ to generate Fig. S12a. Reference [4] highlighted the existence of two FS regimes: the relation $\sigma_i \sim \mu_i^\beta$ with $\beta \approx 1/2$ is evident for small-$\mu_i$ apps, whereas a larger $\beta$ value ($\beta \approx 0.85$) occurs for large-$\mu_i$ apps. Figure S12b shows the corresponding FS plot for the simulated results from the recent-information, long-memory model of Fig. 1g,h of the main text. Clearly, the plot is qualitatively similar to that of the data. In particular, it shows scaling regimes with FS exponents of $\beta \approx 1/2$ for low-$\mu_i$ (i.e., low popularity) apps and $\beta \approx 1$ for high-$\mu_i$ (i.e., high popularity) apps. We now use our model to further analyze these two regimes. (The possible nature of the transition between these regimes is discussed in Ref. [4].)

As we discuss below, our model reveals that the $\beta \approx 1$ scaling of the large-$\mu_i$ apps is related intimately to the large diurnal oscillations in Facebook user activity. Recall that we represent such oscillations at the population level using the function $F(t)$. In simulations using non-oscillatory versions of $F(t)$, we find that the $\beta = 1/2$ regime extends to much larger values of $\mu_i$, which suggests that the $\beta \approx 1$ regime in Fig. S12 appears because very popular apps exhibit coherent diurnal oscillations in their levels of installation activity. By contrast, small-$\mu_i$ apps receive an average of less than 2 installations per hour, and their $f_i(t)$ time series appear similar to shot noise, for which the FS exponent $\beta = 1/2$ is expected.

In the top row of Fig. S13, we show FS plots for the data and the model but calculating $\mu_i$ and $\sigma_i$ in a slightly different manner from that discussed above. (Recall that we calculated $\mu_i$ as the temporal mean of the increments $f_i(t)$ from $t = t_i$ to the final time $t = 1209$; we calculated the standard deviation $\sigma_i$ similarly.) In Fig. S13, however, we instead begin the temporal averaging at $t = t_i + 168$. (If $t_i + 168 > 1209$, then we drop this point from the plot.) This implies that we
calculate the means and standard deviations only over ages from 1 week onwards, so we neglect
the novelty regimes for most apps. Comparing the top left panel of Fig. S13 with Fig. S12a, we
see that this change in definition of \( \mu_i \) and \( \sigma_i \) does not strongly affect the FS plot of the data.
However, as one can see by comparing the top middle panel of Fig. S13 to Fig. S12b, the model
results clearly are impacted by ignoring the novelty regime in calculating \( \mu_i \) and \( \sigma_i \). This arises
from the relatively small fluctuations in the model for very popular apps. For example, the panels
in the bottom row show the \( f_i(t) \) time series for the app ‘What’s your stripper name?’ (see panel
3 of Fig. S3). In the data (bottom left panel), the \( f_i(t) \) time series decays slowly with the age
of the app. However, the model does not reproduce this decay (see the bottom middle panel),
as it instead has a broadly unchanging envelope of \( f_i(t) \) values in the post-novelty regime, and
the fluctuations are due mainly to the aggregate activity level \( F(t) \) that is input to the model.
These fluctuations clearly give the main contribution to the standard deviation \( \sigma_i \) in our revised
definition. Indeed, the diurnal variations are inherited directly from the \( F(t) \) function, and these
fluctuations have the same order of magnitude as the mean: see, in particular, the function \( \psi(t) \)
in Fig. S2. This implies that \( \sigma_i \sim \mu_i \) in this case, which yields an FS scaling exponent of \( \beta = 1 \).

We generate the third panel in each row of Fig. S13 using a further modification of our model:
we replace the total activity function \( F(t) \) that we input to the model with the linear growth
function \( A(t) \) from Fig. S2. This revised model has a total installation activity that grows linearly
in time, but it does not experience the system-wide diurnal variations of the data. We still copy
the \( H = 168 \) hour history window for newly-launched apps directly from the data (see the bottom
right panel). This introduces some residual 24-hour variation, but it is much less prominent than
in the model before modification. The resulting post-novelty standard deviations \( \sigma_i \) for popular
apps are much smaller than in the unmodified model. Moreover, they scale as $\sigma_i \sim \mu_i^{1/2}$ for a much larger range of $\mu_i$ values. (Compare the top right panel to the top middle panel.) We conclude that the high-$\mu_i$ scaling of $\beta \approx 1$ is connected intimately with diurnal variations in the activity levels of Facebook users.

**References**

[1] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. I. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489:295–298, 2012.

[2] S. González-Bailón, J. Borge-Holthoefer, A. Rivero, and Y. Moreno. The dynamics of protest recruitment through an online network. *Scientific Reports*, 1:197, 2011.

[3] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.-L. Barabási, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. Van Alstyne. Computational social science. *Science*, 323(5915):721–723, 2009.

[4] J.-P. Onnela and F. Reed-Tsochas. Spontaneous emergence of social influence in online systems. *Proceedings of the National Academy of Sciences of the United States of America*, 107(43):18375–18380, 2010.

[5] R. A. Bentley, M. Earls, and M. J. O’Brien. *I’ll Have What She’s Having: Mapping Social Behavior*. MIT Press, 2011.

[6] C. R. Shalizi and A. C. Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods and Research*, 40:211–239, 2011.

[7] H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42:425–440, 1955.

[8] A. De Vany. *Hollywood Economics: How Extreme Uncertainty Shapes the Film Industry*. Routledge, 2003.

[9] G. U. Yule. A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FRS. *Philosophical Transactions of the Royal Society of London, Series B.*, 213:21–87, 1925.

[10] W. J. Ewens. *Mathematical Population Genetics: I. Theoretical Introduction*. Springer, 2004.

[11] S. Redner. How popular is your paper? An empirical study of the citation distribution. *The European Physical Journal B*, 4(2):131–134, 1998.

[12] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 695–704. ACM, 2011.
[13] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone’s an influencer: quantifying influence on Twitter. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pages 65–74. ACM, 2011.

[14] K. Lerman, R. Ghosh, and T. Surachawala. Social contagion: An empirical study of information spread on Digg and Twitter follower graphs. arXiv preprint arXiv:1202.3162, 2012.

[15] D. J. de Solla Price. A general theory of bibliometric and other cumulative advantage processes. Journal of the American Society for Information Science, 27:292–306, 1976.

[16] M. V. Simkin and V. P. Roychowdhury. Re-inventing Willis. Physics Reports, 502(1):1–35, 2011.

[17] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. Science, 286(5439):509–512, 1999.

[18] R. A. Bentley, M. W. Hahn, and S. J. Shennan. Random drift and culture change. Proceedings of the Royal Society of London. Series B: Biological Sciences, 271(1547):1443–1450, 2004.

[19] R. A. Bentley, P. Ormerod, and M. Batty. Evolving social influence in large populations. Behavioral Ecology and Sociobiology, 65(3):537–546, 2011.

[20] G. Szabo and B. A. Huberman. Predicting the popularity of online content. Communications of the ACM, 53(8):80–88, 2010.

[21] C. Cattuto, V. Loreto, and L. Pietronero. Semiotic dynamics and collaborative tagging. Proceedings of the National Academy of Sciences of the United States of America, 104(5):1461–1464, 2007.

[22] A.-L. Barabási. Bursts: The Hidden Patterns Behind Everything We Do, from Your E-mail to Bloody Crusades. Dutton Adult, 2010.

[23] S. Fortunato, A. Flammini, and F. Menczer. Scale-free network growth by ranking. Physical Review Letters, 96(21):218701, 2006.

[24] T. S. Evans and A. D. K. Plato. Exact solution for the time evolution of network rewiring models. Physical Review E, 75(5):056101, 2007.

[25] T. E. Harris. The Theory of Branching Processes. Dover Publications, 2002.

[26] S. Zapperi, K. B. Lauritsen, and H. E. Stanley. Self-organized branching processes: mean-field theory for avalanches. Physical Review Letters, 75(22):4071–4074, 1995.

[27] C. Adami and J. Chu. Critical and near-critical branching processes. Physical Review E, 66(1):011907, 2002.

[28] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. Physical Review E, 64(2):026118, 2001.
[29] K. I. Goh, D. S. Lee, B. Kahng, and D. Kim. Sandpile on scale-free networks. *Physical Review Letters*, 91(14):148701, 2003.

[30] G. Ver Steeg, R. Ghosh, and K. Lerman. What stops social epidemics? In *Proceedings of the 5th International Conference on Weblogs and Social Media*, 2011.

[31] J. L. Iribarren and E. Moro. Impact of human activity patterns on the dynamics of information diffusion. *Physical Review Letters*, 103(3):38702, 2009.

[32] J. L. Iribarren and E. Moro. Branching dynamics of viral information spreading. *Physical Review E*, 84(4):046116, 2011.

[33] A. Vazquez, B. Racz, A. Lukacs, and A. L. Barabási. Impact of non-Poissonian activity patterns on spreading processes. *Physical Review Letters*, 98(15):158702, 2007.

[34] A. Zeng, S. Gualdi, M. Medo, and Y.-C. Zhang. Trend prediction in temporal bipartite networks: the case of Movielens, Netflix, and Digg. *arXiv preprint arXiv:1302.3101*, 2013.

[35] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. N. Amaral. A Poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences of the United States of America*, 105(47):18153–18158, 2008.

[36] M. Karsai, K. Kaski, A. L. Barabási, and J. Kertész. Universal features of correlated bursty behaviour. *Scientific Reports*, 2(397), 2012.

[37] P. Bak. *How Nature Works: The Science of Self-Organized Criticality*. Springer, 1999.

[38] M. Beguerisse-Díaz, M. A. Porter, and J.-P. Onnela. Competition for popularity in bipartite networks. *Chaos*, 20:043101, 2010.

[39] J. P. Gleeson, J. A. Ward, K. P. O’Sullivan, and W. T. Lee. Competition-induced criticality in a model of meme popularity. *arXiv preprint arXiv:1305.4328*, 2013.

[40] L. Weng, A. Flammini, A. Vespignani, and F. Menczer. Competition among memes in a world with limited attention. *Scientific Reports*, 2:335, 2012.

[41] J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani. Characterizing and modeling the dynamics of online popularity. *Physical Review Letters*, 105(15):158701, 2010.

[42] T. S. Evans and A. Giometto. Turnover rate of popularity charts in neutral models. *arXiv preprint arXiv:1105.4044*, 2011.

[43] M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, 2006.