Paying Incumbents and Customers to Enter an Industry: Buying Downloads*

Timothy Bresnahan, Xing Li and Pai-Ling Yin†

February 24, 2016

Abstract

Success breeds success in many mass market industries, as well known products gain further consumer acceptance because of their visibility. However, new products must struggle to gain consumer's scarce attention and initiate that virtuous cycle. The newest mass market industry, mobile apps, has these features. Success among apps is highly concentrated, in part because the "top apps lists" recommend apps based on past success as measured by downloads. Consequently, in order to introduce themselves to users, new app developers attempt to gain a position on the top app lists by "buying downloads," i.e., paying a user to download the app onto her device. We leverage a private dataset of one platform for buying downloads and identify the return from this investment. \$100 invested will improve the ranking by 2.2%. To understand the investment rationale for buying downloads, we build a model that accommodates (1) the impact of buying downloads on top list rank, (2) optimal investment in buying downloads, (3) an empirical distinction between app diffusion generated by bought downloads and diffusion from organic downlads, and (4) a rich set of app-specific heterogeneities. We quantify the app-specific structural coefficients by estimating the model using time-series ranking positions of 2,306 free iOS apps. We find the median value of one organic download is 70% of the cost of buying one download, implying a huge marginal cost of buying downloads. App developers lose money during the initial days after release. The coefficients are correlated with ex-post quality, measured by user ratings, but uncorrelated with ex-ante observed app characteristics, suggesting that developers face a great deal of ex-ante uncertainty about the outcome for their apps when they enter the market. We then employ our model to estimate the diffusion delay resulting from the visibility problem in the mobile app industry.

*This research project is based on data collection and analysis over a wide range of data sources. We are grateful to a number of research assistants who have worked on those datasets, gathered industry information, and joined us in industry interviews. Special thanks to Tobias Batton, Tavi Carare, Christopher Correa, Pedro M. Gardete, Wes Hartmann, Shibo Li, Qiaowei Shen, Ping Tu, and seminar participants at the FCC, Stanford University, and Peking University. We are also very grateful to the many industry participants who have shared their time and expertise with us. The usual disclaimers apply.

†Bresnahan: Stanford University, 579 Serra Mall, Stanford, CA 94305 (email: tbres@stanford.edu); Li: Stanford University, 579 Serra Mall, Stanford, CA 94305 (email: xingli@stanford.edu); Yin: Stanford University, 366 Galvez St., Stanford, CA 94305 (email: pyin@stanford.edu)

1 Introduction

The recent explosion in mobile applications (apps) for the iPhone and Android smartphone has created an entry problem for developers: how does an app get noticed out of the million of apps available on each platform? The importance of visibility is exacerbated by two forces that concentrate success into only a few apps. First, many apps target the consumer mass market entertainment, and thus share the same conditions that generate "blockbuster" industries (e.g., movies, music, etc.). Second, the only distribution channel for iPhone apps is through the iTunes App Store, and Google Play is the primary channel for Android. The top lists of apps published by these stores provide one of the most important mechanisms for users to learn about the existence of apps. A top list system makes a successful app more visible, and thus reinforces its success.

Entry is further challenged by nonconvex sunk costs of entry: larger costs are incurred in earlier stages of product diffusion, rather than in proportion to market penetration. Entry for a new app involves gaining enough visibility to gain future visibility. Not surprisingly, our discussions with developers, venture capitalists, and a myriad of firms that have arisen to serve the mobile apps ecosystem have revealed that the largest cost facing developers is the effort that must be spent on marketing, and marketing expenditure is highest during the initial days immediately after the launch of an app.

We find that potential entrants have imperfect forecasts of their apps' demand. A steady stream of new apps appear in the app store and incur those initial marketing costs in the attempt to initiate the virtuous cycle of visibility begetting more visibility. Most of these apps disappear from the top lists within days. The ex-ante expected costs of failure are a second source of nonconvex sunk entry costs.

In many media industries, entrants pay incumbents and/or distributors to gain initial exposure. The modern mobile app industry, with access to big data methods, targeted advertising, and other tools, has industrialized this process. Entering apps "buy downloads" from a brokerage platform such as Tapjoy or Conjectur. These platform firms then pay existing, successful app developers to display an ad for the entering app and also pay mobile app users to download the entering app. Users are paid with in-app products (e.g. moving to the next level in a game app) or even gift cards for off-line retailers. From the entering app's perspective, this is the most effective and important form of advertising, leading directly to a download and thus to a boost in app ranking in the top app lists published daily on iTunes App Store.[1] Thus, an interesting feature of the nonconvex sunk entry costs is that they blend initial negative prices to users with payments from entrants to incumbents. The incumbents may not be in direct product market competition with the entrants, but they are certainly in competition for users' attention.

The welfare implications of buying downloads are complex. Buying downloads mixes informative advertising, as the user is getting a kind of free trial, with dis-informative advertising, as the new app appears higher on the top list as if it were being freely chosen by more consumers. The buying downloads system is a partial offset to the attention bottleneck created by a top list system. However, the form of that partial offset involves paying incumbents.

While we will not be able to disentangle the welfare economics in this context, we will be able to address a number of important positive questions about entry. Does the buying downloads system actually impact top list rankings, and if so, by how much, and at what price? Does the top list system create a barrier to entry, and if so, for which apps? Who benefits from buying downloads? What is the cost imposed by the visibility problem on app adoption?

Our first empirical analysis uses previously unexamined data from a large platform for buying downloads. We find that the return from a \$100 investment in buying downloads will improve the app's ranking position by 2.2% the following day. This nontrivial impact of buying downloads on top ranking positions provides a partial explanation for why entrants make the investment.

In our second empirical analysis, we use historical ranking data for 2,306 iOS apps, following

[1] Advertising campaigns for new apps outside the mobile world itself are quite rare. For most successful mobile apps, the most important source of income is ads for other mobile apps.

each app for the first 300 days after its initial release. We propose a modified Bass diffusion model that incorporates all elements of buying downloads, organic downloads, and visibility effects. We choose the Bass model because of its parsimonious specification and high empirical relevance to the diffusion of new apps. In our model, right after launching the mobile apps, app developers invest in buying downloads intensively to get their apps on top list and achieve a high adoption rate, in order to take off earlier. After taking off, app developers will stop buying downloads, and their app will revert to its organic diffusion pattern.

Applying the model to the data, we quantify the app-specific structural coefficients, including the value per organic download, visibility effect, word-of-mouth effect and market size. We find that the median value of one organic downloads is 70% the cost of buying one download, implying a huge marginal cost of buying downloads. App developers are losing money during the initial days after release. These coefficients are correlated with ex-post quality, measured by users' rating, but uncorrelated with ex-ante observed app characteristics, such as app category, suggesting that developers face a great deal of ex-ante uncertainty about the outcome for their apps when they enter the market. We then employ our model to estimate the diffusion delay resulting from the visibility problem in the iTunes App Store.

1.1 Related Literature

To the best of our knowledge, our paper is the first to quantify the return on investment in advertising expenditure of buying downloads in the mobile app industry. With the explosion in growth of the mobile industry, a solid understanding of digital marketing and advertising is becoming more important, both in academics and in practice.

Methodologically, the challenge of measuring advertising effects is overcome in two ways in the literature: experimental study and natural experiments.[2] Our paper joins the second stream,

[2] Research using controlled experiments includes Lodish et al. (1995); Simester et al. (2009); Bertrand et al. (2010); Sahni (2013); Lewis and Reiley (2014); Blake et al. (2015); Lewis and Rao (2015), while research using natural experiments include Milyo and Waldfogel (1999); Hartmann and Klapper (2012); Lewis and Reiley (2013); Stephens-

using the exogenous design of a platform for buying downloads to evaluate the return from buying downloads as an advertising mechanism.

Due to the data availability, we use rankings as proxies for sales and performance. (Waldfogel, 2012). The ranking position also indicates the endowment of a visibility effect that has a positive feedback on future sales. (Sorensen, 2007). In the mobile app industry, both Carare (2012) and Engstrom and Forsell (2014) use rankings to study the existence visibility effects. In this paper, we endogenize the visibility effects on the diffusion pattern for new apps.

To capture the diffusion of a mobile app, we build on the Bass diffusion model (Bass, 1969, 2004) and its variants and applications. We model buying downloads as a strategic variable that affects the coefficient of innovation, which is similar to the monopolist's advertising decision to affect diffusion in Horsky and Simon (1983). Other benchmark models incorporate ad-like strategic variables in the Bass diffusion process in different ways,[3] but there is no consensus in the literature on the performance of these methods.[4] We use the framework of Horsky and Simon (1983) to model buying downloads because it captures the mobile app industry in a more relevant way.

Although the Bass model is often used to capture the diffusion of new product categories, we use the Bass model to capture the diffusion of a single mobile app product.[5] It is worth mentioning that our paper is among the first to study the diffusion of digital products through digital media, which is expected to be more prevalent in future research. (Rangaswamy and Gupta, 2000).

Finally, we are joining a small but growing literature on the mobile app industry. Many researchers cover different aspects of the mobile industry, including demand for apps (Ghose and Han, 2014), developer's platform choices (Bresnahan, Orsini, and Yin, 2014), innovation strategy (Yin, Davis, and Muzyrya, 2014), and value creation in the whole industry (Bresnahan, Davis, and

Davidowitz et al. (2015)

[3]Such as Kalish (1985); Simon and Sebastian (1987); Dockner and Jørgensen (1988); Bass et al. (1994); Krishnan and Jain (2006)

[4]Horsky and Simon (1983); Mahajan and Muller (1986); Krishnan and Jain (2006)

[5]Givon et al. (1995) captures the diffusion of software, and Hahn et al. (1994); Vakratsas and Kolsarici (2008) capture the diffusion of new drugs.

Yin, 2014). Our paper is among the first to quantify the effect of buying downloads and evaluate the effect of visibility on app diffusion.

The rest of this paper is structured as follows: the next section discusses relevant details of the mobile app industry for our analysis. Section 3 quantifies the return on buying downloads from private data of one such platform. Section 4 proposes a modified Bass model to quantify the effect of buying downloads and top rankings. Section 5 estimates the model using market data and demonstrate important implications, and section 6 concludes.

2 Industry Background

There are various ways that developers can monetize a mobile app. Some app developers sell their apps to consumers, although this is used by less than 10% of the apps. Others adopt "freemium" models or earn money from "in-app purchases." They can also offer free apps and earn money from selling ad space in their apps. Finally, they may also, like many other startups, seek to be a takeover target for another firm.

There are no obviously correct metrics for sales in this industry. Downloads, consumer usage, and in-app purchases are all candidate measures. One common feature for all monetization strategies and sales measures is that they rely heavily on their consumer base. Similar to other industries, mobile app developers earn from their current consumers directly (selling apps or in-app purchases) or indirectly (selling ad space), or nurturing their consumer base to get a high acquisition bid.

Due to the limitation of screen space on smartphones and an explosive increase in mobile apps, it is difficult to make apps noticed even by a small group of people. In order to attract enough consumers, app developers have to invest heavily to advertise their apps. They may rely on conventional ads such as TV and newspaper, or digital media such as tech blogs. Compared to conventional markets, app developers have a new tool, which is to directly purchase downloads from end consumers, or, "buy downloads".

There are different ways to buy downloads. Developers can plant a button in other apps, encouraging users from other apps to click and download. They can do better targeting of users through Facebook Mobile App Ads. They can also buy downloads through platforms such as Tapjoy and Conjectur. Tapjoy offers players in-game coins in return for downloads; Conjectur directly rewards users with gift cards from Amazon, Ebay, and even iTunes for downloads.

Most, if not all, of the incentivized downloads from Tapjoy or Conjectur are valueless. They are usually downloaded by a pool of irrelevant downloaders who are either game players or people that are willing to trade time for money through click-and-install. These downloaders will never seriously use the app nor buy any in-app-purchase. They may remove the app from their mobile phone immediately after installing and acquiring the incentive. The closest counterpart to the mobile app industry's incentivized download system is "payola" in the music industry, where successful artists are paid by newcomers to open before their acts.

The rationale for buying downloads comes from the existence of a top list. Every day, Apple issues rankings for top 500 free, paid and grossing apps. Although the exact algorithms for rankings on these lists are not publicized by Apple, industry wisdom suggests they are heavily weighted towards daily downloads. App developers invest to buy valueless downloads simply in order to appear on a top list and get noticed by potential consumers. Without the existence of top ranking lists, app developers would not buy these downloads.

3 Return to Buying Incentivized Downloads

This section employs a commercial dataset to examine the impact of buying downloads on rankings. This is a distinct dataset from the one we use for model estimates below. This dataset has detailed observations on buying downloads for a modest number of apps.

3.1 Data

The commercial data come from one large platform for buying downloads (referred to as "T-Wall" henceforth). The data cover the period from August 29, 2014 to March 9, 2015, a total of 193 days. In that period, 2,537 apps had an open campaign to buy downloads through T-Wall for at least one day.

The business model for T-Wall is a download auction. Each app submits one campaign at a time. The campaign includes the reward a downloader will be paid for downloading the app and the starting and ending time of the campaign. The rewards are in "tokens" that can be converted to very cash-like instruments, such as Amazon gift cards.

Each day, T-Wall displays the rewards for all apps with an active campaign in descending order: the app with highest reward is on the top, followed by the app with second-highest reward. We show the downloader-facing interface of the platform in Figure 1. Potential dowloaders choose apps to earn the posted rewards. Each downloader has her own cutoff due to her opportunity cost and time budget. The variety in cutoffs leads to a higher rate of downloads for those apps that have posted higher reward relative to other apps. App developers pay only after downloaders download their apps.

We link the data on these campaigns to other sources using the unique app id in the iTunes store. First, we link to the iTunes App Store top 500 rankings for free apps and for paid apps. These rankings are available daily. Other performance data, such as total downloads, are closely held information. Earlier studies with more complete data (including private data) on smaller samples of apps have found an approximately linear relationship between ln(Ranking) and ln(downloads). Of the 2,537 T-Wall apps, 344 appeared on the top rankings during the 193 days. It is this subsample of apps with both rankings data and bought downloads data that we use to quantify the return to buying downloads. Summary statistics of all variables in our linked dataset are presented in Table 1.

Our analysis is at the app-day level, and we assume that the reward offered by other apps is independent of the reward from the focal app. This creates exogenous variation in the expenditure for buying downloads by a particular app through variation in the bidding of other apps. We assume that this variation can be used to identify the return to buying downloads. Our main identification assumption is that the number of installs and total amount rewarded to consumers is exogenous to time-varying, app-specific unobservables, or demand shocks, that may affect the ranking position. There are two reasons the design of T-Wall guarantees our identification assumption. First, the attractiveness of each post is determined by the reward value set ex-ante. Second, the conversion rate depends on the entry of other apps on T-Wall.

We use the total amount of rewards paid to downloaders per app-day as a measure of buying downloads. The expenditure by the app developer is approximately 3 times this amount.[6] To study impact, a precise measure of the amount received by downloaders is preferable to a measure of the amount spent by developers. Among all 344 apps across 193 days, the average amount rewarded per day is \$39.32, with a standard deviation of \$243.00. In about 75% of the app-days, the total amount rewarded is zero, meaning zero downloads.

In order to measure the performance of these apps, we use their ranking positions in the top 500 free or paid apps on iTunes App Store. In our sample, 28% of the 66,392 app-days (344 apps on 193 days) appear on a top 500 list. On the other days, their rankings are below 500 and censored at 500.[7] Conditional on being listed, the average ranking is 227.

We also link to our app questionnaire.[8] The app questionnaire is a detailed survey undertaken by a team of our research assistants. They download the app and use it in order to answer a long list of questions. We rely on a subset of these questions, including whether the app is a corporate app and whether this app monetizes by selling downloads. Corporate apps are defined as an app developed by firms whose main businesses are not on the mobile platform. Bank of America, the New York Times, Facebook and Yelp are corporate apps, whereas Angry Birds, Cleverbot, and Zite

[6]This estimate is based on discussions with industry experts.
[7]We use the sign convention of the industry that 1 is a "high" ranking and 500 is a "lower" ranking.
[8]Discussed in detail in Bresnahan-Davis-Yin (2014).

are not. In our dataset, we have identified 19 corporate apps.[9] If we detect that an app offers an incentive to download other apps, we record that app as *selling* downloads. Other than developer-specific characteristics, we also include app characteristics, i.e. "paid app" and "game" that are collected from the iTunes App Store.

3.2 Empirical Strategy

We use the logarithm of rankings as a dependent variable for the following two reasons. First, the difference in performance between apps of rank 1 and rank 5 is likely to be larger than that between apps ranking 301 and 305. Second, previous researchers document a log-log relation between rankings and downloads.

Since rankings are right censored, we employ the Tobit censored regression model as follows

$$\begin{aligned} \log Ranking^*_{i,t+1} &= \alpha \cdot AmountRewarded_{it} + X_{it}\beta + c_i + d_t + \varepsilon_{it} \\ Ranking_{it} &= \min\left(Ranking^*_{it}, 500\right) \end{aligned}$$

where $Ranking^*_{it}$ represents the actual ranking for app i at day t, $AmountRewarded$ is the total money value of rewards received by downloaders of app i on day t, X_{it} includes other variables, including the age of the app, c_i is an app-specific fixed effect, and d_t is a time fixed effect.

3.3 Impact of Bought Downloads on Rankings

Estimates are reported in Table 2. Without any controls, we estimate that \$100 in rewards paid to users will improve the ranking position by 13.6% (Column 1).

However, this specification might overestimate the return to buying downloads. App-specific unobservables such as the quality of the app may influence both the expenditure and the ranking.

[9] Economic theory strongly suggests that corporate apps are less likely to buy downloads, so this low rate (19/334) is not surprising.

This will lead to a bias if better apps tend to spend more on buying downloads. We check for this by controlling for app fixed effects. The controls lead to a lower estimate of the return to buying downloads. We estimate the return from \$100 paid in rewards to be a 6.7% change in ranking position (Column 2). We interpret the difference between column 1 and column 2, with its significantly lower estimate, as implying that better apps in our sample tend to invest more in buying downloads.

Other controls do not change this estimate much at all. We include “week since release” fixed effects to control for the life cycle of an app (Column 3), and date fixed effects for some time-varying common shocks to ad behavior such as holiday effects (Column 4).

One concern for our estimation is that app developers may exert efforts other than buying downloads from T-Wall to promote their apps. They can buy downloads from other platforms (our industry interviews tell us most do) or engage in other forms of advertising (our industry interviews tell us most do not). We would expect app developers to launch multiple campaigns in different platforms and media simultaneously. These arguments suggest we might be measuring some mixture of the return to buying downloads with the return to buying other media.

We believe this problem is minor thanks to two elements of our strategy. First, we are using the day-to-day variation in rewards and ranking. Second, day-to-day variation in the amount of customer download rewards is not a strategic variable directly chosen by the app developers. Rather, it is exogenously determined by the entry and bidding of other apps in Token Wall. There is thus no reason to believe that advertising or buying downloads behavior outside T-Wall is confounded nor that it biases our estimates.

To quantify the estimates, consider an app whose ranking would be 500 if it did not buy downloads. Such an app would require more than \$5,800 in rewards paid to downloaders to reach the top 10 ranking.[10] Keeping in mind that the reward amount is the value received by the downloaders, not the amount paid by the developers, and that industry sources estimate the ratio as $1/3$, it would

[10] $(\log(500) - \log(10))/0.67 = 5.83K$

cost an app with an "organic" ranking of 500 more than $17,000 to reach the top 10 for a day. An app with a worse organic ranking would pay correspondingly more.

3.4 Endogeneity

The design of T-Wall, where bidding values are determined ex-ante and position on T-Wall is determined by bids of other apps, guarantees the randomness of the total reward. However, there may still be concerns that the existence of app-specific time-varying demand shocks determine bids at the beginning of each campaign. In order to further control for this concern, we use the number of apps in other categories (logged) as an instrument for the total reward for the following two reasons. First, it is related to the position of apps on T-Wall, because high entry of apps may occupy the top positions and shift down the apps, thus decreasing the total reward. Second, it is orthogonal to demand shocks, even considering the competitive behavior among apps, because we use the count of apps from other categories. In the current study, the category is defined by two dimensions: paid/unpaid and games/non-games, so there are four categories in total.

The result using instrumental variables is reported in column 5 of Table 2. Compared with column 2, the magnitude of estimates is unchanged, while the standard error improves, which implies that this kind of endogeneity is not a huge concern in the current context.

3.5 Apple's Apparent Ranking Algorithm

Many industry figures believe that iOS top rankings are determined by downloads on a single day.[11] Because of the widespread curiosity about Apple's ranking algorithm, some industry figures we interviewed have conducted controlled experiments, buying more downloads for a day or two for selected products and watching the impact on the products' rankings.

Our data support this argument. Figure 2 reports the 95% confidence interval of lag-k estimates

[11] See Bresnahan-Davis-Yin (2014) for more discussion. Many of the same figures believe that Android top rankings are a somewhat slower process, a moving average over the last 8 days.

α_k from the censored regression regression

$$\log Ranking^*_{it} = \sum_{k=1}^{10} \alpha_k \cdot AmountRewarded_{i,t-k} + c_i + d_t + \varepsilon_{it}$$

We can see that the amount rewarded for buying downloads only affects the ranking position on that one day, and there are no further effects (only α_1 differs meaningfully from 0). One can view this as confirming the view of the industry figures about Apple's practices.

An alternative view of this finding is that our argument about exogeneity for our regression is correct. We get the same result as the industry participants' experiments.

Using this data alone, we can only estimate the effect of increasing bought downloads on one day. To learn whether the program of buying downloads succeeds over the coming months, we later turn to other data and to a model of developers' decisions to buy downloads.

3.6 Heterogeneous Effects

There is huge heterogeneity across apps. They serve different functions, they are written by different developers, and they target different consumers. The return may also be different across different apps. On the other hand, since all apps in our sample buy downloads from the same platform, T-Wall, and "downloaders" on T-Wall only care about the reward amount, the return is also likely to be neutral to other app characteristics. We estimate a differential return by interacting the amount rewarded with app-specific characteristics. Table 3 reports the results.

First, we see that corporate apps receive lower returns from spending the same amount of money. There are 19 corporate apps in our sample, and their returns are about half the return of other apps. Either corporate apps have already built a large consumer base and the return is smaller, or they are not worried about budget constraints, so they submit their bid on T-Wall less effectively. Data indicates the second interpretation, as there is no difference in rankings between corporate and non-corporate apps, whereas the amount rewarded by corporate apps is significantly higher than that by non-corporate apps.

Second, we see apps that sell downloads themselves do not have a differential return. In the digital advertising industry, selling downloads means participating in the sell-side of ads network, whereas buying downloads is the buy-side behavior. Our results suggest that these two sides are not highly correlated, since there is no higher or lower return to buying downloads by apps who engage in selling downloads.

Third, there is no differential return for game apps. Since all apps in our data buy downloads from the same platform, T-Wall, only differences in bidding strategy across app developers matter for returns; app-specific categories do not matter.

4 Paying to Enter a Positive-Feedback Industry

The results of the previous section are very suggestive. App developers buy downloads, gaining a positive return in ranking on the top-list. However, most of the bought downloads are not customers of the app. In this section, we first discuss an economic model that rationalizes this behavior. We then identify the *other* implications of this model and check them against a new dataset. Finally, we refine the model into a quantitative model of optimal buying downloads behavior by a forward-looking, profit-maximizing developer.

Our economic model includes three significant elements. The *first* element of our model is that consumer attention is limited so that there is a distinction between a product which is potentially attractive to consumers and a product that consumers know of and seek. The diffusion process which leads to large-volume mass-market demand cannot begin until the product is visible to a large number of potential customers. The *second* element of our model posits that the main mechanisms for consumer learning of products focus attention on products that are already at least partially successful. That which is visible is that which is already being seen. In our industry, the key mechanism is the top-list. Consumer search could have much of the same structure, if search results favor those products that many consumers are likely to download. The *third* element of our model is a mechanism for sellers to "buy" some initial success. In our industry, the most important mechanism for this is literally buying downloads via paying customers to take the product. The second most important mechanism is advertising a new app to the users of an existing (and obviously, advertising-supported) app.

These elements provide an explanation for the existence and importance of payola if they are adequately strong empirically. They also have further implications that we can test empirically:

1. Because consumers learn of products from the top lists and from their acquaintances, the industry has a virtuous cycle at the product level, where success breeds further success.

2. The virtuous cycle is adequately important that all firms would like to have visibility to increase their sales, however, the limit on consumer attention is adequately important that this is not possible.

3. The uncertainty about product quality on the part of consumers is adequately important that the virtuous cycle plays out slowly, e.g. through talking to acquaintances (Bass model or similar).

4. The supply of young apps is dominated by sellers' efforts to achieve consumer visibility, while older apps have either fallen away or passed on to their diffusion paths.

In the remainder of this section, we examine the implications of our key economic assumptions in a number of informal tests. The analysis goes beyond the merely descriptive, however, because some of the implications are novelly implied from our theory, not appearing in earlier investigations. After that, we propose a structural model based on the above economic assumptions that accommodates (1) the impact of buying downloads on top list rank, (2) optimal investment in buying downloads, (3) an empirical distinction between organic diffusion (a la Bass) and bought downloads, and (4) a rich set of app-specific heterogeneities.

4.1 Informal tests of model

4.1.1 App rankings, ratings and questionnaires

Our second data set is based on publicly available information about iPhone apps on the English language iTunes store. We examine both individual apps' listings on the store and the "top free" and "top paid" lists assembled by Apple. The apps listing page gives us information like the app's original release (introduction) date, category, and customer ratings. The top lists gives us daily download rankings data for 500 free and 500 paid apps per day. The rankings data give us information about how much the app was downloaded. This information is comparative – it is a ranking –, censored – we see only the top 500 –, and high frequency – we see the rankings daily.

All of our sample apps were first released after Feb. 10, 2010.[12] All appeared on a top list before May 19, 2014.[13] That restriction means we are looking at a selected sample of comparatively successful apps, which is of course where the behavior we study in this paper will be found, if anywhere.[14] All have customer ratings data.[15] Each app has appeared on the top 500 free or paid list for at least eight (not necessarily consecutive) days between Feb. 10, 2010 and May 19, 2014.[16] These sample restrictions reduce the more than 20,000 apps that have ever appeared on 500 lists in the time period to our working sample of 5,518.[17]

Table 4 reports the summary statistics of all variables in the analysis. For each of these apps, we collected its release (original introduction) date, free/paid status, price (if paid), and category. Since

[12]When we began to record rankings and founded our broad research project.

[13]When we froze the data set for this paper.

[14]Of the approximately 1.4 million apps released by the end of our sample period, approximately 20,000 ever appear on a top list.

[15]"Rankings" are where the app falls on the top list. "Ratings" are based on 1-star to 5-star quality ratings entered by users.

[16]This restriction permits identification of our model, which has app-specific parameters, and also permits calculation of statistics like the correlation of rankings at different times.

[17]In terms of sample size, the most important restriction is appearing on the top lists. The next most important restriction is appearing at least 8 times. The restriction that we can find the app introduction date (release date) leads us to drop some older apps, which may list only the release date of their newest version within our sample period. The restriction that we can find customer ratings on our about the 300th day of the app leads us to drop some flash-in-the-pan apps, which hit the top list but then leave the app store before the 300th day.

the complete list of iTunes store categories is quite long and not very informative, we group categories into five buckets: Games/Entertainment/Sports, Books/News, Photo/Video, Music/Travel, and other.

Customers of iOS apps are encouraged to rate each app after installing it, giving one to five stars. For each of the apps, we capture customers' ratings on a rolling basis.[18] We use the number of ratings (logarithm), average ratings, and coefficient of variation of ratings (to control for app demand heterogeneity). Typically, we use these variables as of the 300th day of the app's existence, whether or not it is on the top list at that point.

For each of the apps, we track the app rankings from its birth to the 300th day after its release, which yields 1,655,400 app-days. On most of these app-days, the app is off the top 500 list. An average app appears on the top list (on-list) on only 63 days (or 21% of the time) during its first 300 days. For the 340,704 app-dates where we observe the top list ranking, the average rank conditional on being on a top list is 229.64.[19]

4.1.2 Hump-shaped ranking pattern after product introduction

We find a hump-shaped pattern for rankings, peaking very rapidly after an app's initial release (Figure 3). Figure 3a plots the probability of being on the top list against days since release. On the release date, 12% of the apps reach the top list. This probability grows dramatically within the first *two weeks*, peaking at 47% in day 12. After that, it drops gradually down to a steady level of about 17% in six months. The same pattern is also observed in the average rankings. (Figure 3b). An average listed app ranks 460 on the release day. This ranking grows to 350 in two weeks, then drops after that.

This very rapid hump is consistent with what industry participants have told us about buying

[18]Due to the limited computing power of our computers and of the app store's servers, we do not obtain ratings for every app every day. Where there are gaps, we fill in with a linear interpolation.

[19]If we had a sample of top-list-days, i.e. with 500 apps from each day, the average app ranking would mechanically be 250.5. By conditioning on apps that appear 8 or more times and our other data filters, we induce a slightly better (lower) average ranking of 229.64.

downloads right after product release. It is also consistent with our examination of the efficacy of buying downloads in the prior section, under the assumption that a substantial fraction of new apps have a marketing budget for buying downloads. Apps buy downloads for a short initial period, then stop buying downloads once they have attained enough visibility to launch their organic diffusion or once they realize they will not be a successful app.

4.1.3 Multiple Humps

One explanation for this hump could be extremely rapid diffusion of apps to their audiences, with the hump being that implied by a standard Bass model with very rapid diffusion. We are, however, quite certain that this is not the appropriate interpretation.

First, most of the downloads of apps generally occurs well after the initial peak period for *new apps* shown in Figure 3. Consider a typical day on the app store, such as June 28, 2013. On that day, we can observe the release date for 848 of 1,000 apps on top 500 free and paid lists. An average app is on 575 days since its release, and a median app is on 338 days. Ninety-seven (of the 1,000) apps were in the first 60 days of their diffusion. Most of the downloads, however, are going to significantly older apps. There must be another peak (at least for successful apps) after the peak at day 12.

Our model has the testable implication that product sales can have two peaks: one associated with buying downloads, and the other later one associated with organic diffusion. Figure 4 provides two examples whose second humps occur within the first year of our data period.

We are not the first to detect a double-humped diffusion pattern. Chandrasekaran and Tellis (2011) documented three explanations: chasms in the adopter segments, business cycles, and technological cycles. The double-humped app diffusion pattern caused by buying downloads is consistent with the first explanation: there are two temporarily segmented groups of customers, and the dip after the first hump is caused by the decrease in the diffusion among the early group of consumers.[20] In our application, buying downloads happens earlier, and the diffusion of organic downloads among real customers happens later.

Detecting these two peaks is key to testing our theory, and it requires two different strategies. The first peak can be seen directly in the data for newly introduced products, since it applies to all apps, both those that will be successful and those that will eventually fail. Also, our framework implies that the *first* peak has the same economic logic for all apps that buy downloads: they are all attempting to obtain a high-visibility spot on the top list. The later peak, once (and if) they are established,is determined by the characteristics of demand for the individual product. So only the *first* peak is most easily seen by examining the early download history of cohorts of new apps, such as in Figure 3.

Accordingly, we define a new variable. An app is *rising* at time t if its ranking is better (i.e. a lower number) at time $t + s$ than at t. Figure 6 plots the mean of this variable, i.e. the probability an app whose ranking is observed at time t is rising, for three different values of s.[21] If we use a seven-day lag to label rising apps, the probability of rising apps declines from 50% at release day, to 12% in day 12, consistent with what we saw above – clearly there is a trough in "rising" which corresponds to the peak in rankings at age 12 days.

However, the fraction rising does not continue to decline after 12 days, as it would if there were only a single peak. The fraction of apps which are rising at time t – i.e., the fraction which are before a peak – is steady in the general neighborhood of 30% from about day 100 for the rest of the first year. For those apps which are successful, a substantial fraction of them have a second peak long, long after the 12th day. Figure 6 empirically motivates our modification of the standard Bass diffusion model to account for buying downloads, i.e., of keeping the organic peak

[20] Other examples include Mahajan et al. (1990); Moore (1991); Goldenberg et al. (2002); Van den Bulte and Joshi (2007); Vakratsas and Kolsarici (2008)

[21] We calculate the probability conditional on ranking being non-censored at time t, i.e, $\Pr(R_{i,t} > R_{i,t+s} \mid R_{i,t} < 500)$ where R means ranking, because we cannot compare two censored rankings. Note that this definition does *not* condition on observing the ranking at time $t + s$. An app that is top-ranked at time t but not at time $t + s$ is not a rising app at t under our definition.

of diffusion which is at a reasonable time (varying across apps) and permitting BDL behavior to drive an earlier peak.

4.1.4 Auto-correlation and persistence of rankings

This subsection provides another piece of evidence for the significance of buying downloads in the early days of app diffusion. We showed that earlier rankings are less predictive of long-run rankings than are later rankings. This pattern would arise if demand for products, organic downloads in our framework, only became known to demanders after a period of time. It would also arrive if suppliers of heterogeneous apps controlled their level of downloads to similar levels at an early stage in order to gain visibility. Both of these stories are consistent with our framework.

To examine this, we look at how on-list status at an earlier time $t = 12$ and a later time $t = 100$ predict future on-list status separately. In Figure 7a we plot, from day 100 after release to day 200, the probability that apps are on-list on that day conditional on four events: on/off list at day 12 (2,517/3,001 apps), and on/off list at day 100(1,130/4,388 apps). We have two observations. First, based on the rankings around day 300, apps that are on-list at day 100 are superior, in the sense of long run performance, to apps that are on-list at day 12. This is also confirmed by fewer on-list apps at day 100 (1,130) compared to 2,517 on-list apps at day 12. This suggests a selection effect. However, if there was only a selection effect, the 4,388 apps that are off-list in day 100 should also be better than 3,001 apps that are off-list in day 12, contrary to our findings. It must therefore be true that some off-list apps in day 12 appear on the top list at day 100, and the probability of such cases is smaller between day 100 and day 200. In other words, day 12 ranking is less predictive of day 200 ranking than day 100 ranking.

To quantify this "predictability", we run the regression

$$y_{i,t+s} = \rho_0 + \rho_{t,s} y_{it} + \varepsilon_{it}$$

for app i at day since release t and $t+s$, where y is measured as a dummy for being on a top list or as the log of ranking. The coefficient $\rho_{t,s}$ measures the correlation in y between day t and day $t+s$. Figure 8 reports the estimates of ρ for different t and s when we use a dummy for being on a top list (Figure 8a) and log of ranking (Figure 8b) as dependent variables. Different lines in Figure 8 plot the estimates for different values of s. It is not surprising to see that the value is decreasing in s, as it becomes more difficult to predict the performance with a longer time lag. For fixed s, the coefficient ρ is unanimously increasing in t. When the y variable is the top list dummy, the 7-day correlation between day 1 and day 8 is as low as 0.4. This number almost doubles in three months, and stays steady at a level of 0.8. A similar pattern can be observed when y is the log of listed ranking.

4.2 Theoretical Model

The empirical patterns discovered in our two datasets, reported in sections 3 and 4.1 above, guide our model of a developer's optimal dynamic investment, paying to enter a positive-feedback industry. Our model is built on top of a standard Bass model of product diffusion (Bass, 1969). We add model elements that capture buying downloads in order to appear on the top-list, an empirical pattern central to our modeling strategy. There is an important distinction between organic customers who choose the product and bought downloaders whose only role is to propel the product onto the top list.

We model the diffusion of organic customers, following Bass, as having an advertising effect p and a word-of-mouth effect q that provides positive feedback.[22] Critically, in our model the advertising effect p only applies to apps that are visible on the top list, so we re-label p as a "visibility effect". In our model, a higher download rate will increase the probability of being on

[22]The large literature has given these effects a number of names: innovation/imitation, external influence/internal influence, and advertising/word-of-mouth. The math of the model is the same for all these names, with q providing a positive feedback effect unlike p. We use the informational names because of our treatment of p as following from the product's visibility on the top list.

the top-list (and thus getting p, the visibility effect). This provides a second element of positive feedback.

That a higher adoption rate increases the probability of being on the top list also provides the motivation for the developer to "buy downloads." In our model, buying downloads is endogenously determined by app developers' dynamic profit-maximizing. The developer's strategy depends on the underlying economic fundamentals just described, the state of its visibility on the top-list, and the state of the app's diffusion.

More specifically, at the beginning of each period t, the app developer decides on the number downloads to buy B_t, to maximize

$$V\left(F_t, d_t\right) = \max_B \left\{\pi \cdot S_t - C(B) + \delta E\left(V\left(F_{t+1}, d_{t+1}\right) | F_t, d_t, B\right)\right\} \tag{1}$$

where the costs of these downloads are borne at once, $C(B_t)$. No revenue is ever received from the bought downloaders B_t.

The benefits are dynamic. At time t, there are S_t customer downloads. Each customer download is associated with net revenue of π.[23] S_t follows a modified Bass model with parameters (p, q, M). Here M is the market potential, p is the coefficient for the visibility effect, and q is the coefficient for the word of mouth effect. Letting F_t, be the cumulative rate of adoption, f_t be the flow rate of adoption, and d_t be a dummy for whether the app is visible on the top list at time t. S_t, the flow of customer choices of the product, follows the Bass law of motion with one change:

[23]More precisely, π is the expected present value of revenue from a customer choosing the product. In our application, the products are free at the point they are chosen by customers, but many of the customers will later make "in app payments" (IAP). Then π is the expected present value of those IAP.

$$f_t = (d_t \cdot p + q \cdot F_t)(1 - F_t) \tag{2}$$

$$S_t = M \cdot f_t \tag{3}$$

$$F_{t+1} = F_t + f_t \tag{4}$$

The rate of adoption in period t is f_t. From (2), the organic downloads come from two sources: the top-list visibility effect $d_t \cdot p$ and the word-of-mouth effect $q \cdot F_t$. Note that the "advertising" – now called "visibility" – flow of customers arises only when the app is on the top list ($d_t = 1$). Other that that, equations (2) to (4) are the standard Bass model. In the early days of app diffusion to paying customers, there is no word-of-mouth effect because no consumers have ever used the app ($F = 0$). Only through pushing the app onto the top list can app developers begin the process of diffusion to customers.

The model is completed by adding the equations determining d, presence on the top list:

$$\log R_t = \beta_{0R} + \beta_R \log(S_t + B) + u_t \tag{5}$$

$$d_{t+1} = I(R_t < \Phi) \tag{6}$$

Ranking position R_t is log-linear in total downloads, which is a mixture of organic downloads S_t and bought downloads B, with some measurement error u_t. The future visibility effect, d_{t+1}, equals one if the ranking lies above some threshold level Φ.

A profit maximizing developer maximizes (1) subject to (2) through (6)

Figure 9 displays the solution to the model for one set of coefficients. The value function is plotted in the top panel; the corresponding policy function is plotted in the bottom panel. The value function is decreasing in F, as the app diffuses. The value with a visibility effect ($d = 1$) is higher than that without a visibility effect ($d = 0$). The difference in values with and without a visibility effect leads the app developer to invest in buying downloads. The difference is decreasing in F, so the policy function in the bottom panel is also decreasing in F. It is easy to show that app developers are willing to invest less in buying downloads when $d_t = 1$. In the example displayed in Figure 9, they will buy zero downloads in that case.

4.3 Empirical model

This subsection proposes an empirical implementation of our model.

The empirical implementation is based both on the logic of the economic model and the available data. We base the model on the iTunes App Store data. Like all other publicly-available datasets, this does not record sales or downloads. We rely on the log-log relation between downloads and rankings and use rankings to proxy downloads. A further advantage of this approach is that it permits easy integration of the top-ranking effects in (5) and (6).

We reverse the sign of rankings in defining our dependent variable to avoid repeated repetition of the sign convention (lower rank is more downloads), and recognize that rankings are censored above at 500. This gives us an approximation of downloads for app i at day t:

$$
\begin{aligned}
y_{it}^* &= \log 500 - \log Ranking_{it}^* \\
y_{it} &= \max\left(y_{it}^*, 0\right)
\end{aligned}
$$

We work with a model in which there are four free parameters and let them vary by app. Three of the four parameters come from the standard Bass, including the coefficient of visibility effect p_i, the coefficient of word-of-mouth effect q_i, and market potential M_i. A fourth parameter, w_i, measures monetization ability. More specifically, we assume a linear cost function, i.e., $C(B) = C \cdot B$, and define the variable $w_i \equiv \pi_i / C$.[24] Furthermore, we normalize both B and the value

[24] We assume revenue per downloads, π_i are app-specific, whereas the marginal cost for buying downloads is common to all apps, stemming from a competitive market for buying downloads.

function V by dividing them by M_i. We define new constants β_0 and β from β_{0R} and β_R, taking our sign change and censoring into account.[25] Finally, to account for the sign change we define a new measurement error term $u_{it} = -u_{it}^R$.

These normalizations yield $w_i = \pi_i/C$ and not π_i or C, which cannot be separately identified without monetary variables. They also normalize all bought or organic download variables (B, S) by M as they cannot be identified without a scale variable. For the same reason, it is easy to show that $(\alpha, \mu_m = E(\log M_i))$ are not jointly identified.

Taking stock of normalizations, we have

$$\begin{aligned} w_i &= \frac{\pi_i}{C} \\ b_{it} &= \frac{B_{it}}{M_i} \\ m_i &= \beta_0 + \beta \log M_i \\ v_{it} &= \frac{V_{it}}{M_i} \end{aligned}$$

The app developer's problem becomes

$$v_i(F_{it}, d_{it}) = \max_b \{w_i f_{it} - b + \delta E(v_i(F_{i,t+1}, d_{i,t+1}) | F_{it}, d_{it}, b)\} \tag{7}$$

[25] Specifically, $\beta = -\beta_R$ to account for the sign change and $\beta_0 = \log(500) - \beta_{0R}$ to account for the sign change and censoring.

such that

$$
\begin{aligned}
f_{it} &= \left(d_{it} \cdot p_i + q_i \cdot F_{it}\right)\left(1 - F_{it}\right) \\
y_{it}^* &= m_i + \beta \cdot \log\left(f_{it} + b_{it}\right) + u_{it} \\
F_{i,t+1} &= F_{it} + f_{it} \\
d_{i,t+1} &= I\left(y_{it}^* \geq \phi\right) \\
&= I\left(\frac{m_i - \phi}{\sigma_u} + \frac{\beta}{\sigma_u} \cdot \log\left(f_{it} + b_{it}\right) + \frac{u_{it}}{\sigma_u} > 0\right)
\end{aligned}
$$

writting app-specific parameters as

$$
\gamma_i = \left(w_i, p_i, q_i, \frac{m_i - \phi}{\sigma_u}\right)
$$

and the policy function, i.e., the decision about buying downloads, is parameterized as

$$
b_{it} = b\left(F_{it}, d_{it}; \gamma_i, \lambda = \frac{\beta}{\sigma_u}\right) \tag{8}
$$

Finally, we have two approaches. In one, we assume the unconditional distribution of the app-specific parameters follows a known parametric form. In the other, we allow a rich specification of the dependence of app-specific coefficients (a one-to-one mapping of γ_i) on regressors as

$$
\alpha_i = \left(\log w_i, \text{logit}(p_i), \text{logit}(q_i), m_i\right)' = \Gamma z_i + \Sigma^{\frac{1}{2}} \epsilon_i
$$

where $\text{logit}(x) = \log(x) - \log(1 - x)$ maps the unit interval to the real line, Γ is a 4-by-k matrix that measures the dependence of app-specific structural parameters on k app-specific observables z_i, including a constant. $\Sigma^{\frac{1}{2}}$ is the Cholesky decomposition of the covariance matrix, and $\epsilon_i \sim N\left(0, I\right)$.

The last three app-specific coefficients, (p_i, q_i, m_i), are taken from the Bass diffusion model.

In the current model, m is a quality measure based on (log) market potential, the maximum number of consumers that will eventually download. The coefficient p relates to the significance of visibility effects, i.e., the rate of customers attracted by appearance on the top list. Finally, the coefficient q measures word-of-mouth effect for diffusion. The model also permits heterogeneity across different apps with respect to all three coefficients.

The ratio between value per organic download and cost per bought download, w_i, is a new coefficient introduced by our modified model. Heterogeneity in this coefficient recognizes variations in monetization potential for different apps. Since a profit-maximizing app developer has monetization as a goal, the estimation of the distribution of w_i and its interaction with app characteristics and app diffusion patterns is of great practical interest.

4.4 Identification

We build our (informal) identification argument upon the standard Bass model with available sales data. The identification leverages the parsimonious specification of (p, q, m). The shape of the Bass curve (over) identifies the two coefficients (p, q) by time to take-off, peak time and duration. Intuitively, high p speeds up the take-off but may not affect the duration length. High q helps all consumers to adopt earlier, thus affecting the duration length. Market potential, m, is identified by the level of the Bass curve.

The identification of the download value/cost coefficient, w, relies on the deviation between the data and standard Bass curve. The standard Bass curve is single-peaked and symmetric in the peak time, whereas our model predicts the possibility of double peaks, one at the beginning and one later, as well as the asymmetry that apps are fast to reach the peak level, but take longer to depreciate. The deviation from the standard Bass model identifies the magnitude of buying download behavior, and thus the value/cost coefficient. All app-specific heterogeneities are identified by observed data within an app.

4.5 Estimation

We use importance sampling (Ackerberg, 2009) to estimate the model, which allows us to calculate the value function iteration only once before searching over parameter spaces. In the current model, the policy function (8) is parameterized by four random coefficients and one common parameter λ. To ease computation, we fix the value of λ at 0.1.[26] Furthermore, we set that the threshold for visibility effects to be top 50, or, $\phi = \log 500 - \log 50$. Finally, the discount factor is set as $\delta = 0.9999$.[27] The parameters of interest are the distributional parameters

$$\theta = \left(\Gamma, \Sigma^{-\frac{1}{2}}, \sigma_u\right)$$

To implement importance sampling, we simulate app-specific coefficients $\tilde{\gamma}_k$ for K = 500 times from a normal distribution $\phi(\cdot|\theta_{IS})$ as

$$\tilde{\gamma}_k = \left(\log w_k, \log\left(\text{logit}(p_k + q_k) - \text{logit}(q_k)\right), \text{logit}(q_k), \frac{m_k - \phi}{\sigma_u}\right)$$

where $\text{logit}(x) = \log(x) - \log(1-x)$. $\tilde{\gamma}_k$ is a one-to-one mapping of app-specific structural parameter γ that takes values on the whole real line and satisfies the restrictions $w > 0$, $p \in (0,1)$, $q \in (0,1)$, and $p + q < 1$. The value function iteration and policy function is calculated on the K simulated coefficients $\tilde{\gamma}_k$ in advance, and the maximum likelihood estimator is equivalent to choosing optimal weights on these K simulated $\tilde{\gamma}_k$. More specifically,

$$\begin{aligned}
\log L &= \frac{1}{N}\sum_i \log\left(\int\left(\prod_t f\left(y_{it}|\alpha_i, \theta\right)\right)\cdot\phi\left(\alpha_i|\theta, z_i\right)\cdot d\alpha_i\right) \\
&= \frac{1}{N}\sum_i \log\left(\frac{1}{K}\sum_k\left(\left(\prod_t f\left(y_{it}|\tilde{\gamma}_k, \sigma_u\right)\right)\cdot\frac{\phi\left(\alpha_k|\theta, z_i\right)}{\phi\left(\tilde{\gamma}_k|\theta_{IS}\right)}\cdot J_k\right)\right)
\end{aligned}$$

[26]This value is set from an estimation of the standard Bass model using the same data set.

[27]Industry discussions suggest that top 50 is a typical target for app success. A daily discount rate of $\delta = 0.9999$ implies an annual interest rate of 4%.

where

$$\begin{aligned}\log f\left(y_{it}|\tilde{\gamma}_k,\sigma_u\right) &= I\left(y_{it}>0\right)\left(\log\phi\left(\frac{y_{it}-m_k-\beta\log\left(f_{ikt}+b_{ikt}\right)}{\sigma_u}\right)-\log\sigma_u\right)\\ &\quad +I\left(y_{it}=0\right)\log\left(1-\Phi\left(\frac{m_k+\beta\log\left(f_{ikt}+b_{ikt}\right)}{\sigma_u}\right)\right)\end{aligned}$$

and the Jacobian

$$J_k=\left|\frac{\partial\alpha_k}{\partial\tilde{\gamma}_k}\right|^{-1}$$

5 Results

We estimate our model on the sample of 2,306 free apps on iTunes App Store described in section (4.1) above.[28] The dependent variable is the ranking of the app each day for the first 300 days since its release. These apps are relatively successful; to be in our sample, they must be on the top list on at least 8 days during these first 300 days. Our substantive interpretation will be limited to apps in this relatively-successful class.

Summary statistics for all variables in this subsample of free apps are reported in the bottom panel of Table 4. We make use of the estimates from these apps to discover the buying downloads behavior and quantify the effect of top rank lists.

5.1 No observed heterogeneity

Estimates of θ without app-specific observed heterogeneity is reported in Table 5. The top of the table presents estimates of Γ, which here has only one row, with Σ and σ_u below. The same estimates, changed into native units (w instead of $\log(w)$, etc.) are reported in Table 6.

Perhaps the most interesting app-specific parameter is w_i, the value of an organic download (as a proportion of cost per bought download). Our estimate of the mean of the distribution of w_i

[28] Estimations for paid apps and all apps are not reported, but the implications remain the same.

across apps is 0.72, and the median is 0.68. These are consistent with what industry figures have told us, that cost per install is about twice the life-time customer value. The distribution of w is log-normal, and it is graphed in Figure 10. For 83% of apps, w_i<1, i.e. the developer is only acting rationally if every investment in a bought download leads, in expectation, to more than one organic download.

Unlike the distribution of w (mild skew) and of m (mean very near median), the distribution of diffusion coefficients p and q are highly skewed, as the median is much closer to zero than the mean. We return to the discussion of app heterogeneity after examining some economic and managerial features of the estimates for the median app.

For the median app reported in Table 6, model simulation (reported in Figure 11) indicates that buying downloads decreases over time and leads to a first peak of total downloads at day 10. The volume of downloads bought becomes almost zero after two months, but by then organic diffusion kicks in. There is a second peak in downloads in this simulation – the peak day of organic downloads arrives on day 464. According to our simulation, before day 37, the flow cost of buying downloads is higher than the flow revenue from organic downloads. The median developer's BDL behavior can only be rationalized as an investment in getting to that second peak. At the median of our estimates, therefore, the economic theory of buying downloads to gain visibility and thus gain organic downloads appears to be the center of the marketing problem facing a developer.

One obvious comparison is to a model in which visibility effects are available for free ($w = \infty$). In that model, diffusion would be much faster, with a peak day of 241 $(\log(q) - \log(p))/(p + q))$ compared to 464 days in our model. This large difference implies that the visibility problem delays the arrival of product maturity by almost one year, even after developers buy downloads to accelerate diffusion.

Magnitudes of heterogeneity, or standard deviations of the random coefficients, are reported in the middle part of Table 5. In our estimation, heterogeneity is large for everything. Since w is a value/cost ratio, and costs per download probably do not vary much in this market, this means that heterogeneous values of organic downloads (i.e., the ability to monetize an organic download) is driving the heterogeneity in w.

Our empirical specification also estimates the correlation for the four app-specific random coefficients, which are also reported in the middle part of Table 5 and the bottom part of Table 6. Value per organic download w is negatively correlated with the shape coefficients of diffusion p and q, which means that apps better at monetization attract a smaller *proportion* of customers due to the visibility effect, and they also have a long diffusion process driven by small coefficients for word-of-mouth effect. The two shape coefficients p and q are positively and highly correlated, so visibility effects and word-of-mouth effects will either both be high or both be low. Finally, the market size m is uncorrelated with the monetization coefficient w and shape coefficients p and q, implying the levels and the shapes of diffusion patterns are quite independent.

The last part of Table 5 reports the performance of our baseline specification. Our estimation of the standard deviation of measurement error σ_u is 1.67, with a log-likelihood value of -210.56. In a simple Tobit model with no regressors to fit the dependent variable (not reported here), the standard deviation of error is estimated to be 2.14, with a log-likelihood value of -239.83. The pseudo R-square of the current model is 39%,[29] and the LR statistics is 58.5 (d.f. = 13),[30] which is significant.

5.2 With observed heterogeneity

Our full model allows heterogeneity across apps to depend on the observed app-specific characteristics. We include three groups of characteristics in our model, based on app ratings, app categories, and developer characteristics. Estimation results are reported in columns (1)-(4) of Table 7. Summary statistics of all the characteristics are reported in Table 4.

We derive the first group of characteristics from user ratings of the app. We include (log)

[29] $1 - \left(\frac{1.67}{2.14}\right)^2 = 0.39$

[30] The current model has 15 parameters, including four means, 10 parameters in the Cholesky decomposition of the covariance matrix, and one σ_u, whereas the simple Tobit model with no regressors has only 2 parameters.

number of ratings, average rating, and coefficient of variation of ratings at day 300.[31] Both the number of ratings and the average measure the quality of an app. The number of ratings is more related to popularity, which may be determined by app quality or other factors such as network effects and competition in the app's product market. Average rating, on the other hand, is mainly driven by app quality as perceived by its own customers. Coefficient of variation of app ratings measures the app's niche-ness.

We begin with discussion of the app-ratings characteristics results, as these are the almost all the results that appear to be estimated with any precision.

From the estimation, we find that app popularity does not necessarily imply better monetization, whereas a higher average rating does. To quantify the estimates, one standard deviation change in average rating (0.72) will increase the value per downloads by 1.4%. As for the market size, number of ratings is associated with a larger market, whereas apps with higher average ratings are more likely to be small in market size (Column 4). The app developer is facing a tradeoff between monetization (high w) and penetration (high m). Apps with higher average ratings are more successful at monetization, but may be reaching a more targeted and favorable customer base to do so. We characterize this app as having a niche market.

The niche apps are also reflected in the negative correlation between the dispersion of ratings, which is measured by the coefficient of variation (standard deviation divided by mean), and market size in Column 4. Controlling for the average rating and number of ratings, an app with a higher dispersion of ratings implies a nich-market app: only a specific segment of customers like it. On the other hand, if the customer ratings are quite uniform, the app likely appeals to the mass market, since many customers in the market are equally attracted to the app.

[31] Chintagunta, Gopinath, and Venkataraman (2010) use average rating, number of ratings, and precision (inverse of variance) of ratings to estimate the effect of ratings on box revenue in the movie industry, and they find that only the average rating matters. Our context is different from theirs in two respects. First, types of reviewers in these two industries differ. In the movie industry, reviewers are a smaller group of customers who tend to be more professional, so their views tends to be more uniform. The insignficant effect of reviewer dispersion on box revenue may due to this lack of variation. Second, their paper makes use of variations within a movie, whereas our paper makes use of variations across apps.

All three characteristics that are derived from customer ratings have negative and significant impacts on the shape coefficients (p and q) of the diffusion pattern, which explains the high correlation between these two in Table 6.

The second group of characteristics relates to the categories of apps. Although ex-post metrics based on customer ratings have significant explanatory power in value per downloads, market size, and diffusion, ex-ante categories of apps have less explanatory power. The only exception is for the game apps, whose developers reach a relatively smaller set (compared to, for example, productivity and social network apps) while exploiting them more. None of the other app categories have any explanatory power on the heterogeneity of monetization, market size, or shape of diffusion.

The third group of characteristics relates to app developers. We identify two special groups: corporate apps (made by developers who do not primarily operate in the mobile app space) and apps that sell downloads themselves. The only significant finding is that apps that selling downloads tend to have a smaller visibility effect, i.e., they are not discovered by the top-list appearance. These apps are likely already quite visible given their ability to sell downloads to other apps.

App-specific characteristics have poor prediction power despite the significant correlation from the ex-post metrics from customer reviews. This can also be seen in the estimation of σ_u and the log-likelihood. Although the model contains 40 more parameters, the standard deviation of the structural error σ_u does not change at all, which implies that these app-specific observables do not explain the ranking variation. The LR statistic between models with and without observed heterogeneity is 1.29 (d.f.=40), which is not significant at all.

The fact that adding observed heterogeneity only produces marginal differences in estimation reflects the high uncertainty that app developers face when they enter the market. In particular, performance cannot be predicted by ex-ante app-specific characteristics.

5.3 Effects of Top List on App Diffusion and Proliferation

The limited customer attention in the mobile app industry creates a visibility problem that prevents the diffusion of mobile apps. The existence of a top-list helps to mitigate this problem, although it imposes entry costs which are paid to incumbents and customers. In order to quantify the visibility problem and the effect of the top list, we propose two counterfactual scenarios.

The first counterfactual scenario is the standard Bass case. In this case, there is no visibility problem, and all mobile apps get the flow of downloads p for free every day. In our model, this corresponds to the case where $w = \infty$. The second counterfactual scenario is the case with the visibility problem, but the absence of a top list, so there is no way for the new app to get noticed. In our model, this corresponds to the case where the cost of buying downloads is infinity, or $w = 0$. This case is too extreme, because the model simulation then implies that most apps will have to wait for more than 10 years before diffusion takes off, so we instead consider a scenario where the cost of buying downloads is doubled, (i.e., w is only half the size).

To measure the app diffusion, we use the "half-life" for each app (the day when half of the diffusion takes place) characterized by app-specific coefficients (w, p, q, m). For the Bass model, the "half-life" is calculated by solving for t from the equation

$$\frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p} e^{-(p+q)t}} = 0.5$$

For the case of the current model and a model with doubled cost, we calculate the expected half-life for each app by simulating its diffusion process over a length of 2000 days.

Figure 12 reports the cumulative distribution function for the half-life across heterogenous apps simulated from Table 5. First, the industry has a severe visibility problem, because the distance between the Bass case and current model is huge. For the 10% of apps with fastest diffusion, the half-life is 9 days without invisibility problem (Bass), but 99 days in the current model. The two half-lives become 122 and 1,054 days for the top 20% apps. Roughly, the half-lives of mobile apps will be shortened by 90% without the visibility problem, which implies a much faster diffusion. Secondly, the differences are marginal after doubling the cost of buying downloads, a proxy for the situation where no top list exists to mitigate the visibility problem. The half-life is 108 days for the 10% of apps with the fastest distribution and 1,137 days for the top 20% of app under this scenario. Compared to the half-lives of 99 and 1,054 day in the current model, doubling the cost of buying downloads will increase the half-lives by 10%.

The existence of visibility effects has improved app diffusion by changing the half-lives from more than 10 years to three months for the top 10% of apps. How much is the app developer willing to pay for a further-improved top-list? To quantify this question, we calculate the life-time value per organic download under the model and two counterfactual scenarios, assuming the cost per bought download is $1.[32] In the Bass model, the profit per download is calculated as

$$w \cdot \sum_{t=1}^{\infty} \delta^{t-1} \left(\frac{(p+q)^2}{p} \cdot \frac{e^{-(p+q)t}}{\left(1 + \frac{q}{p} e^{-(p+q)t}\right)^2} \right)$$

whereas in the other two cases, the profit is already calculated in the value function in (7) as

$$v(0, 0; \alpha)$$

Simulation implies that the average revenue per download is $0.233 in the Bass case, and average profit per download is $0.172 under current model, and $0.169 in case where cost per download bought is doubled. First, an average developer is willing to pay at most $0.061 to improve the efficiency of top-list, because we do not count for any of the cost in calculating the revenue in the Bass case. It is documented that the life-time downloads for an average app is less than 1,000, so the average app is willing to pay less than $60 dollars even in the most favorable scenario. Secondly, it is interesting to compare the simulated profits in the current model, and in

[32] The (log) market size is identified up to a constant, so we do not have a clear inference from our model about the total downloads.

the model with doubled cost. Theoretically, we expect that the value function is non-decreasing in the cost parameter, as in (1), $V(pr, C)$ is non-decreasing in C. In the limit case with $C = 0$, the value function converges to the Bass revenue. From the simulation, we only observe a tiny loss in profits when doubling the cost, and there is room for larger improvement ($0.061) if the cost can be decreased even more.

Our counterfactual analysis suggests that the current top-list system has improved the diffusion of apps given the visibility problem in the mobile app industry, but it points to a glaring problem in app diffusion stemming from the visibility problem in the first place, and suggests that much more can be done to speed up the diffusion of apps if the visibility problem can be further reduced.

6 Conclusion

We study mobile apps as a new mass market industry where consumer attention is limited but important to app success. Current success will make the app more visible, stimulate word of mouth communication, and breed further success. In order to gain this virtuous cycle of success breeding success , app developers invest in buying downloads as a form of advertising.

We make use of the private data and design of one platform to identify the return to buying downloads. $100 invested will improves the ranking by 2.2%. Apps with higher rankings are also more aggressive in buying downloads. We find a different return level for established apps.

To rationalize the level of buying downloads and quantify the top list virtuous cycle, we propose a modified Bass model with endogenous buying downloads. In our model, apps get a constant mass of new consumers, captured by the rate of innovation in the standard Bass, only if they are endowed with visibility effects. App developers buy downloads in order to get this visibility effect. The empirical analog of the model treats "organic downloads" and "actual downloads" differently, and deals with data censoring problem created by top 500 list. A rich specification of app-specific heterogeneity is allowed, including the monetization ability, app quality, and diffusion coefficients.

We estimate the model on a set of apps from iTunes App Store and find a great deal of heterogeneity in apps' monetization, visibility effects, word of mouth effects and app quality. Mass market apps benefit the most from buying downloads, while niche apps and apps with alternative distribution channels benefit relatively less. App-specific characteristics cannot explain the heterogeneity, and app developers thus facing huge uncertainty when making entry investment decisions. We show that while the top list does mitigate the visibility problem in the app market somewhat, there is a huge gain to app diffusion that could be made if the visibility problem could be eliminated.

References

Ackerberg, D. A. (2009). A new use of importance sampling to reduce computational burden in simulation estimation. *Quantitative marketing and economics 7*(4), 343–376.

Bass, F. M. (1969). A new product growth model for consumer durables. *Management Sciences 15*(5), 215–227.

Bass, F. M. (2004). Comments on “a new product growth for model consumer durables the bass model”. *Management science 50*(12), 1833–1840.

Bass, F. M., T. V. Krishnan, and D. C. Jain (1994). Why the bass model fits without decision variables. *Marketing science 13*(3), 203–223.

Bertrand, M., D. Karlan, S. Mullainathan, E. Shafir, and J. Zinman (2010). What’s advertising content worth? evidence from a consumer credit marketing field experiment. *The Quarterly Journal of Economics 125*(1), 263–306.

Blake, T., C. Nosko, and S. Tadelis (2015). Consumer heterogeneity and paid search effectiveness: A large-scale field experiment. *Econometrica 83*(1), 155–174.

Bresnahan, T., J. Orsini, and P.-L. Yin (2014). Platform choice by mobile app developers. NBER Working Paper.

Bresnahan, T. F., J. P. Davis, and P.-L. Yin (2014). Economic value creation in mobile applications. In *The Changing Frontier: Rethinking Science and Innovation Policy*. University of Chicago Press.

Carare, O. (2012). The impact of bestseller rank on demand: Evidence from the app market. *International Economic Review 53*(3), 717–742.

Chandrasekaran, D. and G. J. Tellis (2011). Getting a grip on the saddle: Chasms or cycles? *Journal of Marketing 75*(4), 21–34.

Chintagunta, P. K., S. Gopinath, and S. Venkataraman (2010). The effects of online user reviews on movie box office performance: Accounting for sequential rollout and aggregation across local markets. *Marketing Science 29*(5), 944–957.

Dockner, E. and S. Jørgensen (1988). Optimal advertising policies for diffusion models of new product innovation in monopolistic situations. *Management Science 34*(1), 119–130.

Engstrom, P. and E. Forsell (2014). Demand effects of consumers’ stated and revealed preferences. Working Paper.

Ghose, A. and S. P. Han (2014). Estimating demand for mobile applications in the new economy. *Management Science 60*(6), 1470–1488.

Givon, M., V. Mahajan, and E. Muller (1995). Software piracy: Estimation of lost sales and the impact on software diffusion. *The Journal of Marketing*, 29–37.

Goldenberg, J., B. Libai, and E. Muller (2002). Riding the saddle: How cross-market communications can create a major slump in sales. *Journal of Marketing 66*(2), 1–16.

Hahn, M., S. Park, L. Krishnamurthi, and A. A. Zoltners (1994). Analysis of new product diffusion using a four-segment trial-repeat model. *Marketing Science 13*(3), 224–247.

Hartmann, W. R. and D. Klapper (2012). Super bowl ads. Working Paper.

Horsky, D. and L. S. Simon (1983). Advertising and the diffusion of new products. *Marketing Science 2*(1), 1–17.

Kalish, S. (1985). A new product adoption model with price, advertising, and uncertainty. *Management science 31*(12), 1569–1585.

Krishnan, T. V. and D. C. Jain (2006). Optimal dynamic advertising policy for new products. *Management Science 52*(12), 1957–1969.

Lewis, R. A. and J. M. Rao (2015). The unfavorable economics of measuring the returns to advertising*. *The Quarterly Journal of Economics*, qjv023.

Lewis, R. A. and D. H. Reiley (2013). Down-to-the-minute effects of super bowl advertising on online search behavior. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 639–656. ACM.

Lewis, R. A. and D. H. Reiley (2014). Online ads and offline sales: measuring the effect of retail advertising via a controlled experiment on yahoo! *Quantitative Marketing and Economics 12*(3), 235–266.

Lodish, L. M., M. M. Abraham, J. Livelsberger, B. Lubetkin, B. Richardson, and M. E. Stevens (1995). A summary of fifty-five in-market experimental estimates of the long-term effect of tv advertising. *Marketing Science 14*(3_supplement), G133–G140.

Mahajan, V. and E. Muller (1986). Advertising pulsing policies for generating awareness for new products. *Marketing Science 5*(2), 89–106.

Mahajan, V., E. Muller, and F. M. Bass (1990). New product diffusion models in marketing: A review and directions for research. *The journal of marketing 54*(1), 1–26.

Milyo, J. and J. Waldfogel (1999). The effect of price advertising on prices: Evidence in the wake of 44 liquormart. *American Economic Review*, 1081–1096.

Moore, G. A. (1991). Crossing the chasm: Marketing and selling technology products to mainstream consumers.

Rangaswamy, A. and S. Gupta (2000). Innovation adoption and diffusion in the digital environment: some research opportunities. *New Product Diffusion Models*, 75–96.

Sahni, N. (2013). Effect of temporal spacing between advertising exposures: evidence from online field experiments. Working Paper.

Simester, D., Y. J. Hu, E. Brynjolfsson, and E. T. Anderson (2009). Dynamics of retail advertising: Evidence from a field experiment. *Economic Inquiry 47*(3), 482–499.

Simon, H. and K.-H. Sebastian (1987). Diffusion and advertising: the german telephone campaign. *Management Science 33*(4), 451–466.

Sorensen, A. T. (2007). Bestseller lists and product variety. *The journal of industrial economics 55*(4), 715–738.

Stephens-Davidowitz, S., H. Varian, and M. D. Smith (2015). Super returns to super bowl ads?

Vakratsas, D. and C. Kolsarici (2008). A dual-market diffusion model for a new prescription pharmaceutical. *International Journal of Research in Marketing 25*(4), 282–293.

Van den Bulte, C. and Y. V. Joshi (2007). New product diffusion with influentials and imitators. *Marketing Science 26*(3), 400–421.

Waldfogel, J. (2012). Copyright protection, technological change, and the quality of new products: Evidence from recorded music since napster. *Journal of Law and Economics 55*(4), 715–740.

Yin, P.-L., J. P. Davis, and Y. Muzyrya (2014). Entrepreneurial innovation: Killer apps in the iphone ecosystem. In *American Economic Review, Papers and Proceedings, Forthcoming*.

2,306

Table 1: Summary Statistics

	N	mean	sd	min	max
Total Reward (USD)	66392	39.32	242.96	0.00	4265.64
Positive Reward	66392	0.25	0.43	0.00	1.00
On Top Ranking List	66392	0.28	0.45	0.00	1.00
Listed Ranking	18432	227.08	138.82	1.00	500.00
Corporate	344	0.06	0.23	0.00	1.00
Selling Downloads	344	0.13	0.33	0.00	1.00
Paid	344	0.04	0.20	0.00	1.00
Games	344	0.60	0.49	0.00	1.00

Notes: Data for 344 apps over 193 days (08/29/2014 - 03/09/2015) are collected from different sources. The first block shows information for 66932 app-days (344 $\times$193). Total reward paid and the positive reward dummy are from T-Wall. The dummy for appearing on top rankings list is from the iTunes App Store. The second block shows ranking data from the iTunes App Store conditional on it being observed. The third block shows app-level statistics. The dummies for corporate apps and for apps that themselves sell downloads are from our app questionnaire. Dummies for paid app and game are collected from the iTunes App Store. Rankings are right-censored at 500.

Table 2: Impact of Buying Downloads on Rankings

Dependent variable is $\log Ranking_{t+1}$					
	Tobit				IV Tobit
	(1)	(2)	(3)	(4)	(5)
Total Reward (K USD)	-1.36***	-0.67***	-0.66***	-0.67***	-0.67***
	(0.20)	(0.14)	(0.10)	(0.10)	(0.03)
App FE	No	Yes	Yes	Yes	Yes
Week since release FE	No	No	Yes	Yes	No
Date FE	No	No	No	Yes	No
Observations	66048	66048	66048	66048	66048
Pseudo R^2	0.02	0.47	0.49	0.49	

Standard errors clustered at the level of apps.

* p<0.10, ** p<0.05, *** p<0.01

Notes: This table estimates the impact on top-ranking position of the total dollar value of rewards to downloaders spent on T-Wall. For data sources and definitions, see Table 1. Column 1-4 are Tobits censored at 500, and Column 5 uses the number of apps in other categories as an IV for total reward. App FE (fixed effects), weeks since release FE, and Date FE are described in text.

Table 3: Heterogeneous Effects

Tobit, dependent variable is $\log Ranking_{t+1}$			
	(1)	(2)	(3)
Total Reward (K USD)	-0.70***	-0.70***	-0.82***
	(0.11)	(0.11)	(0.13)
Total Reward (K USD) $\times$ Corporate	0.34*	0.31**	0.40**
	(0.18)	(0.13)	(0.17)
Total Reward (K USD) $\times$ Selling Downloads		0.05	0.11
		(0.17)	(0.18)
Total Reward (K USD) $\times$ Game			0.13
			(0.17)
App FE	Yes	Yes	Yes
Week since release FE	Yes	Yes	Yes
Calendar date fe	Yes	Yes	Yes
Observations	66048	66048	66048
Pseudo R^2	0.49	0.49	0.49

Standard errors clustered at the level of apps.

* p<0.10, ** p<0.05, *** p<0.01

Notes: This table reports the heterogeneous impact of the total amount rewarded to "downloaders" from T-Wall. For data sources and definitions, see Table 1. App FE (fixed effects), weeks since release FE, and Date FE are described in text.

Table 4: Summary Statistics

	N	Mean	Std. dev.	Min	Max
All apps:					
Paid	5,518	0.50	0.50	0.00	1.00
Price	5,518	1.06	1.97	0.00	49.99
Ln Num rate at day 300	5,518	6.60	1.94	1.61	13.97
Avg. rate at day 300	5,518	3.84	0.70	1.00	5.00
Cv. rate at day 300	5,518	0.37	0.16	0.00	0.84
Game,entertain,sports	5,518	0.64	0.48	0.00	1.00
Books,news	5,518	0.01	0.12	0.00	1.00
Photo,video	5,518	0.04	0.19	0.00	1.00
Music,travel	5,518	0.03	0.18	0.00	1.00
Corporate	5,518	0.01	0.09	0.00	1.00
Selling downloads	5,518	0.25	0.43	0.00	1.00
Days since release	1,655,400	150.50	86.60	1.00	300.00
On top list	1,655,400	0.21	0.40	0.00	1.00
Ranking	340,704	229.64	148.08	1.00	500.00
Free apps:					
Ln Num rate at day 300	2,306	7.44	1.93	1.79	13.38
Avg. rate at day 300	2,306	3.76	0.73	1.26	5.00
Cv. rate at day 300	2,306	0.39	0.17	0.00	0.84
Game,entertain,sports	2,306	0.66	0.47	0.00	1.00
Books,news	2,306	0.01	0.10	0.00	1.00
Photo,video	2,306	0.03	0.18	0.00	1.00
Music,travel	2,306	0.04	0.19	0.00	1.00
Corporate	2,306	0.02	0.12	0.00	1.00
Selling downloads	2,306	0.44	0.50	0.00	1.00
Days since release	691,800	150.50	86.60	1.00	300.00
On top list	691,800	0.23	0.42	0.00	1.00
Ranking	156,323	231.39	149.01	1.00	500.00

Notes: The top panel reports the summary statics for the whole sample, whereas the bottom part for the subsample of free apps. The 5,518(2,306) apps are all those that (the subset of free apps that) satisfy the criteria laid out in text. The 1,655,400(691,800) app days are all the days $5518 * 300(2,306 * 300)$ in our sample period for all apps (free apps) in our sample. The sample for Ranking data of 340,704(156,323 for Free) is censored because we only observe Ranking<501.

Table 5: Estimation Without App-Specific Observables - Free Apps

	$\log w$	logit(p)	logit(q)	m
		Γ		
Const	-0.39***	-9.56***	-7.13***	3.51***
	(0.03)	(0.47)	(0.15)	(0.32)
		Σ		
Std. Deviation	0.36	4.00	2.06	2.62
Correlation				
$\log w$	1.00	-0.51	-0.36	-0.09
$g(p)$		1.00	0.73	-0.04
$g(q)$			1.00	0.06
m				1.00
σ_u		1.67 (0.00066)		
Number of Apps		2306		
Log Likelihood		-210.56		

Notes: This table reports simulated maximum likelihood estimates with Importance Sampling. App-specific heterogeneities are specified as $\gamma_i = (\log w_i, \text{logit}(p_i), \text{logit}(q_i), m_i) = \Gamma z_i + \Sigma\epsilon_i$, where $\text{logit}(x) = \log x - \log(1-x)$ maps the unit interval to the real line. In this table, z_i includes only a constant term.

Table 6: Distribution of App-specific Parameters - Free Apps

	w	p	q	m
Mean	0.72	0.02	0.01	3.52
Median	0.68	7.3e-05	8.2e-04	3.51
Std. Deviation	0.26	0.08	0.02	2.63
		Correlation		
w	1.00	-0.18	-0.15	-0.09
p		1.00	0.48	-0.03
q			1.00	0.04
m				1.00

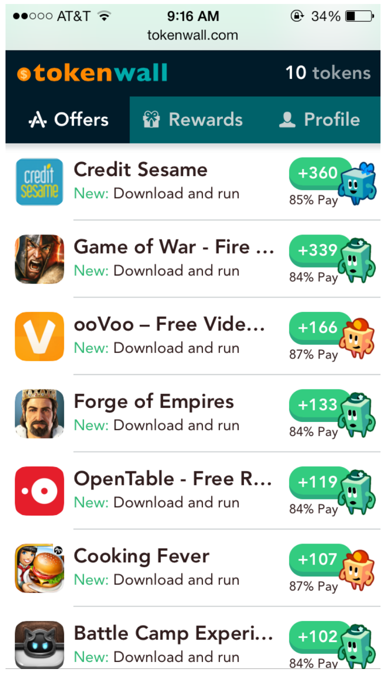
Notes: This table reports the distribution of app-specific coefficients implied by the estimates in Table 5.

Table 7: Estimation With App-Specific Observables - Free Apps

	$\log w$	logit(p)	logit(q)	m
	(1)	(2)	(3)	(4)
		Γ		
Const	-0.54***	-1.81***	-0.78***	5.27***
	(0.06)	(0.36)	(0.28)	(0.23)
Rating Avg.	0.02*	-1.17***	-0.97***	-0.85***
	(0.01)	(0.08)	(0.06)	(0.05)
Rating CV.	0.07	-4.16***	-3.15***	-1.63***
	(0.09)	(0.57)	(0.44)	(0.37)
Rating Ln Number	0.01	-0.26***	-0.19***	0.11***
	(0.01)	(0.04)	(0.03)	(0.02)
Game,Entertain,Sports	0.07*	-0.34	-0.40**	-0.46***
	(0.04)	(0.25)	(0.19)	(0.16)
Books,News	0.04	0.48	0.15	0.20
	(0.21)	(1.24)	(0.94)	(0.82)
Photo,Video	-0.02	-0.13	-0.06	0.00
	(0.11)	(0.83)	(0.49)	(0.43)
Music,Travel	-0.01	0.09	0.03	-0.10
	(0.11)	(0.58)	(0.41)	(0.41)
Corporate	-0.08	-0.57	-0.17	0.30
	(0.19)	(0.76)	(0.61)	(0.62)
Selling Downloads	-0.00	-0.48**	-0.22	-0.05
	(0.04)	(0.24)	(0.18)	(0.15)
		Σ		
Std. Deviation	0.41	2.11	1.61	1.91
Correlation				
$\log w$	1.00	-0.09	0.05	-0.01
$g(p)$		1.00	0.43	-0.16
$g(q)$			1.00	-0.05
m				1.00
σ_u		1.67 (0.00057)		
Number of apps		2306		
Log Likelihood		-209.92		

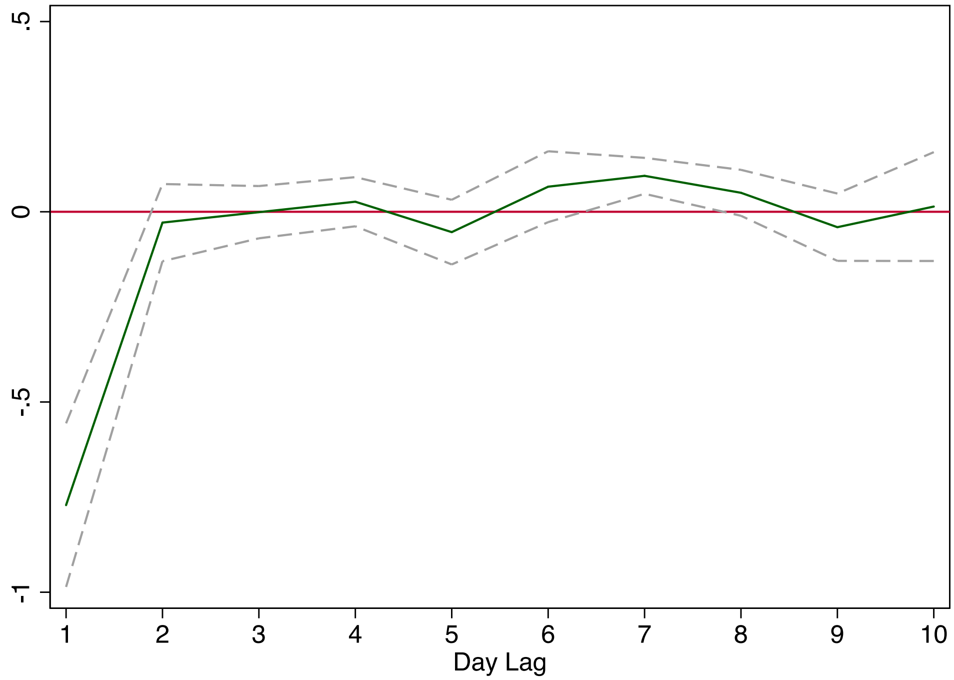
Notes: This table reports simulated maximum likelihood estimates with Importance Sampling. App-specific heterogeneities are specified as $\gamma_i = (\log w_i, \text{logit}(p_i), \text{logit}(q_i), m_i) = \Gamma z_i + \Sigma\epsilon_i$, where $\text{logit}(x) = \log x - \log(1-x)$ maps the unit interval to the real line.z_i includes all variables listed on the left.

Figure 1: Buying Downloads



Notes: This is the downloader (end user) interface of T-Wall on a mobile phone.

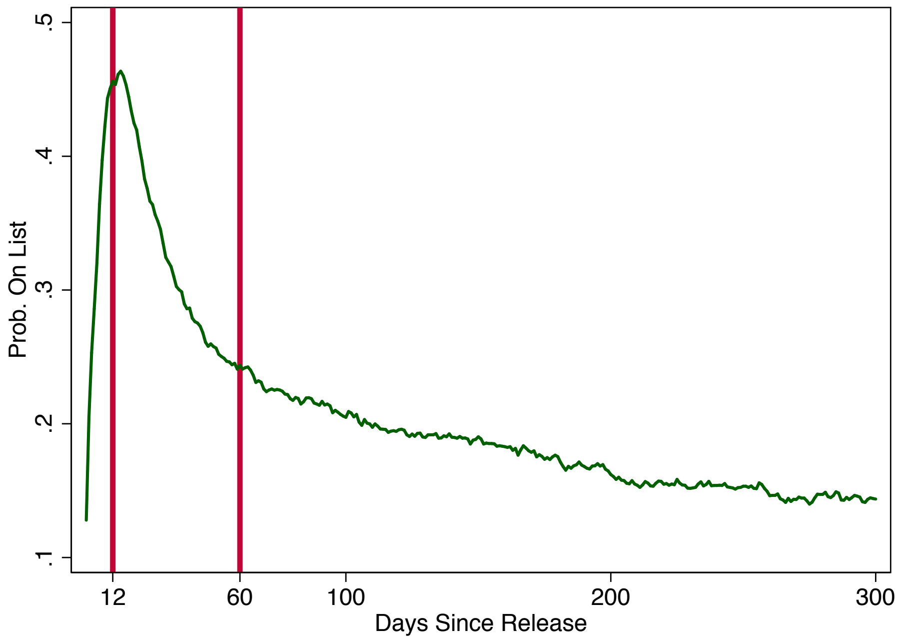
Figure 2: Lag Effects of Buying Downloads



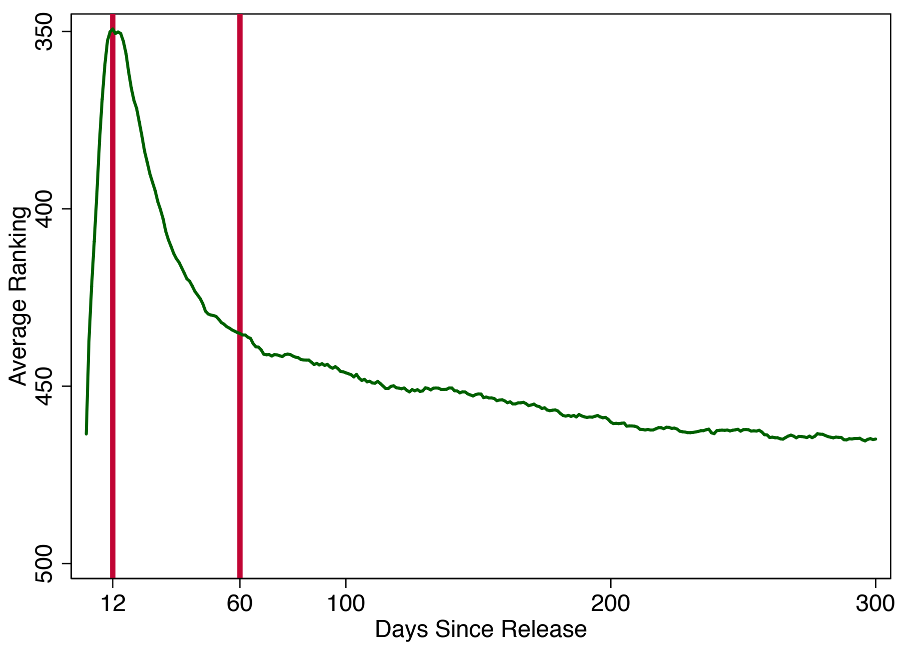
Notes: This figure plots the time-varying estimates of the return to buying downloads. More specifically, it plots the 95% confidence interval for all estimates α_k from the censored regression $\log Ranking^*_{it} = \sum_{k=1}^{10} \alpha_k \cdot AmountRewarded_{i,t-k} + c_i + d_t + \varepsilon_{it}$. For data sources and definitions, see Table 1. App FE (fixed effects), weeks since release FE, and Date FE are described in text.

Figure 3: Hump-shaped pattern

(a) Probability On List



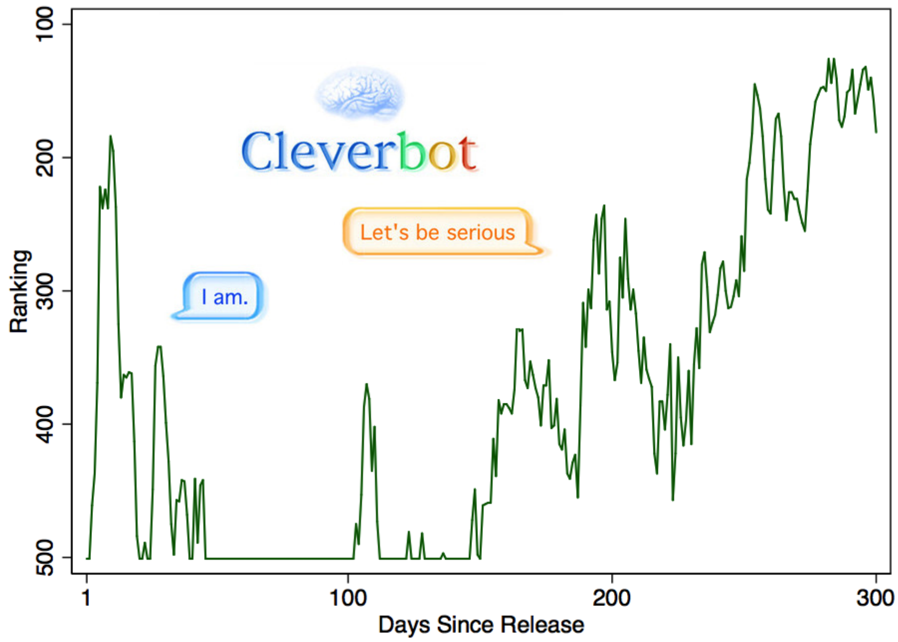
(b) Average Ranking



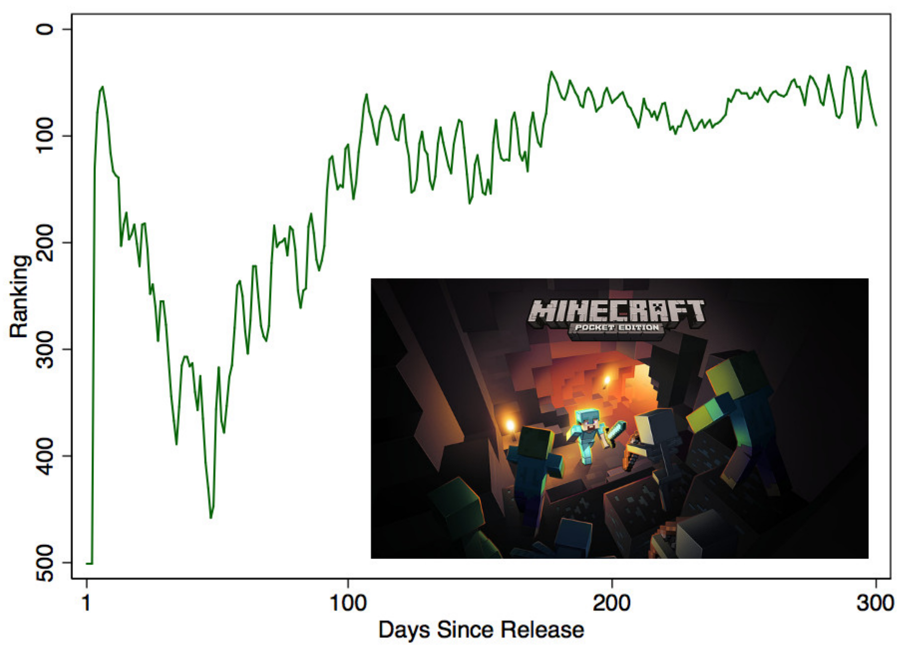
Notes: These figures plot (a) the probability of appearing on the top list and (b) the average ranking | appearance for all apps by day since release. The sample for part (a) is the 5,518 iOS apps and 1,655,400 app-days as described in text. The sample for part (b) is censored because ranking data are only observed if the ranking is <501, leaving 340,704 app-days.

Figure 4: Examples of double peaks

(a) Cleverbot



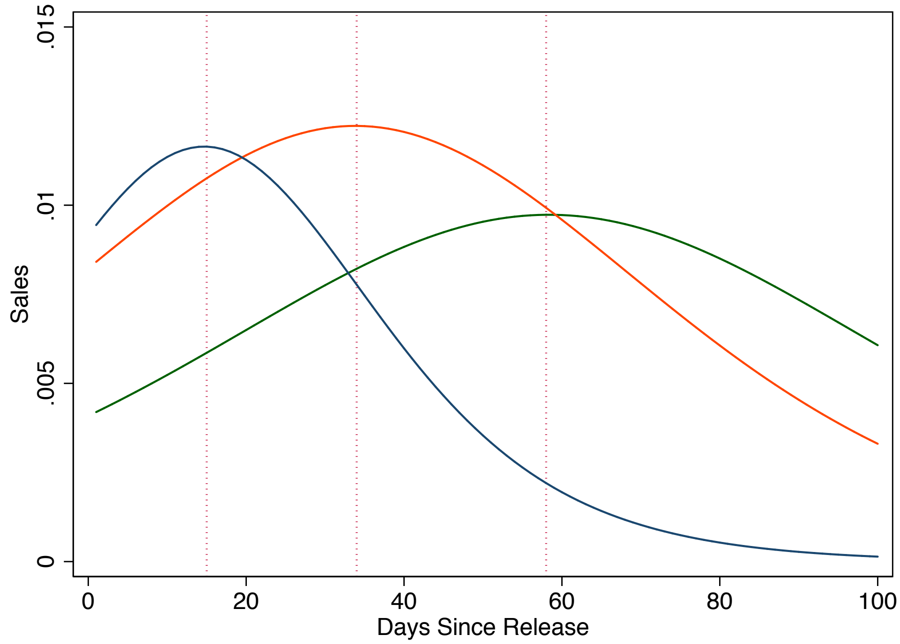
(b) Minecraft



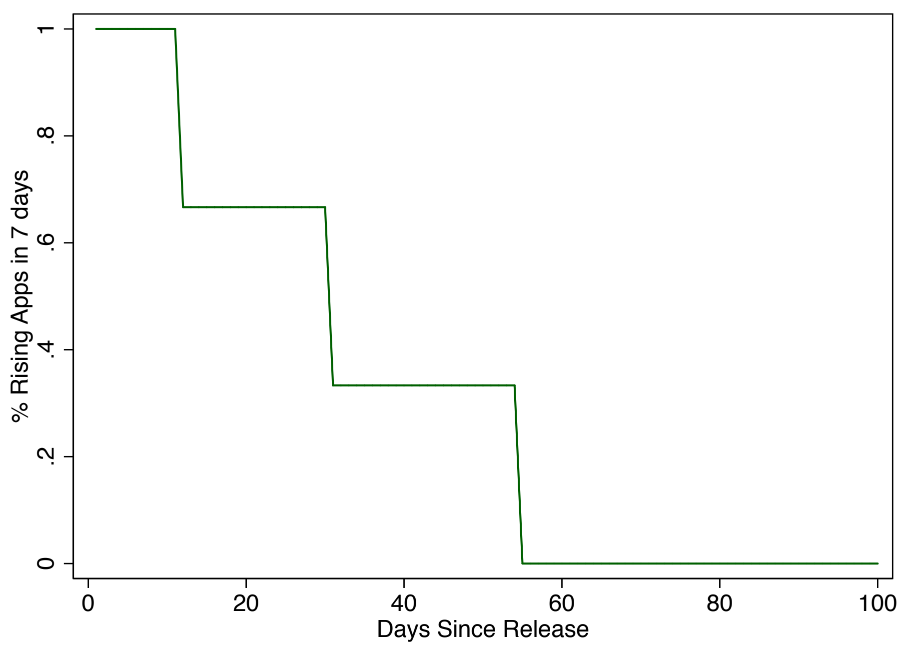
Notes: This figure plots the ranking patterns for two apps: Cleverbot and Minecraft, that exhibit two humps within the first year since initial release. Censored observations for Cleverbot are shown as R=500.

Figure 5: Single Peak Implied from Bass

(a) Bass Curve

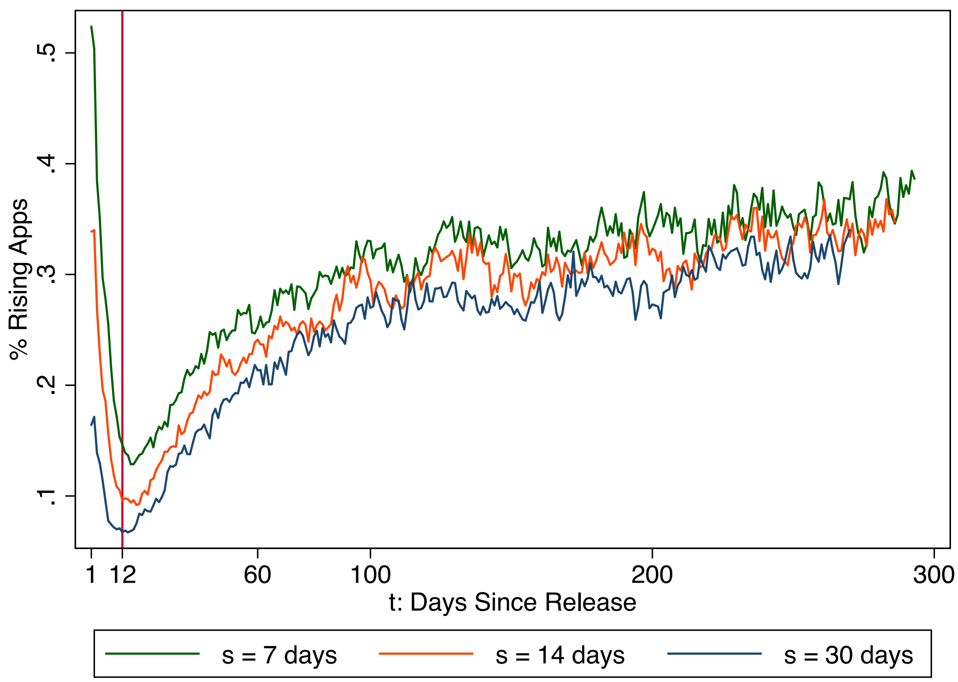


(b) Probability of Rising Apps



Notes: This figure illustrates the implication of single peak for rising apps from the standard Bass model. The top panel plots the Bass diffusion curve of three apps, and the bottom panel plots the probability of rising apps among the above three, by days since release. In the assumption of standard Bass, the probability of rising app is decreasing in days since release.

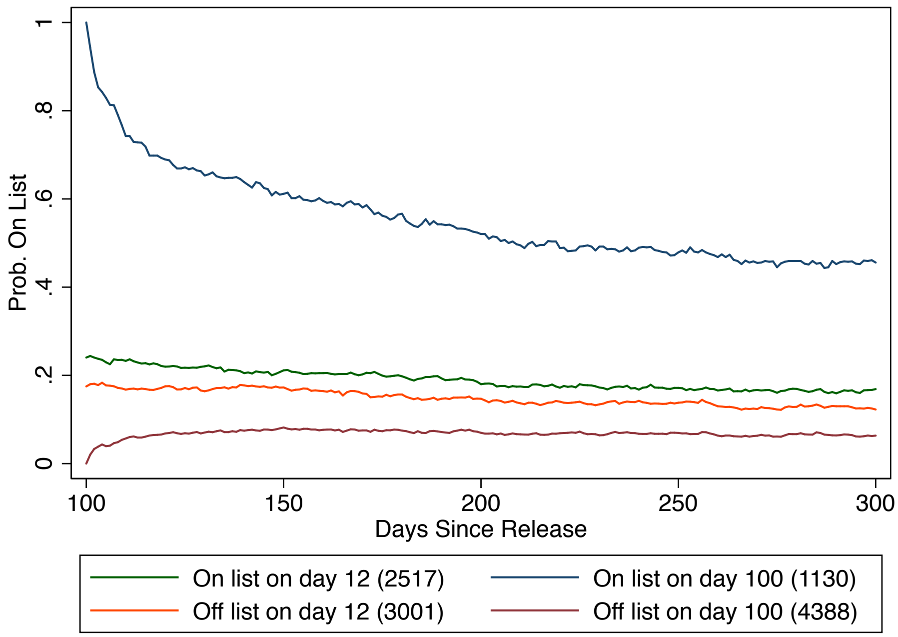
Figure 6: Share of Apps with Improving Rankings



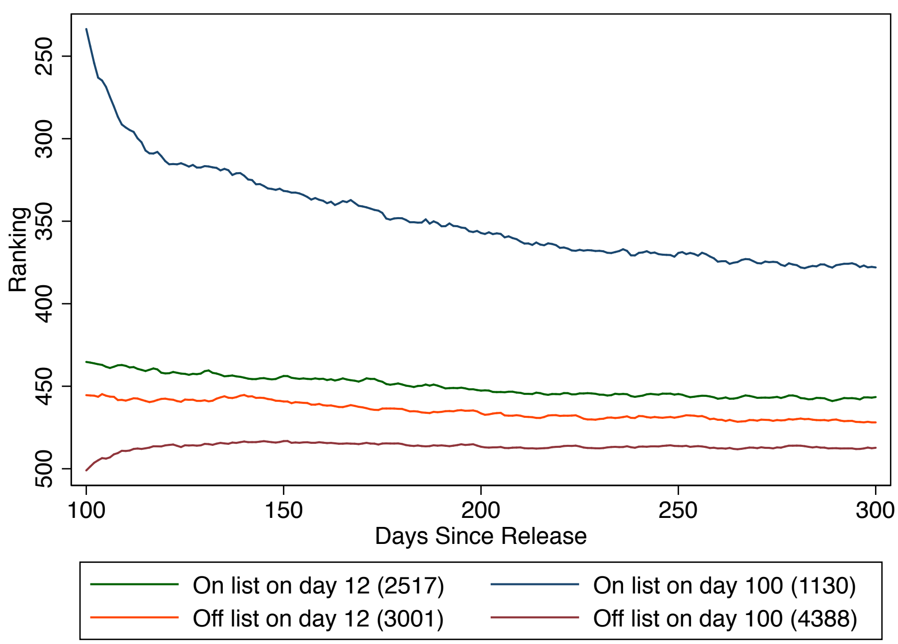
Notes: This figure plots the proportion of rising apps by days since release conditional on being uncensored, i.e, $\Pr(Ranking_{i,t} > Ranking_{i,t+s} | Ranking_{i,t} < 500)$. The x-axis measures the value of t, and different lines represent different values of s. Sample is app days where apps' ranking is observed at t and $t + s < 300$.

Figure 7: Heterogenous Evolution of Ranking

(a) On List



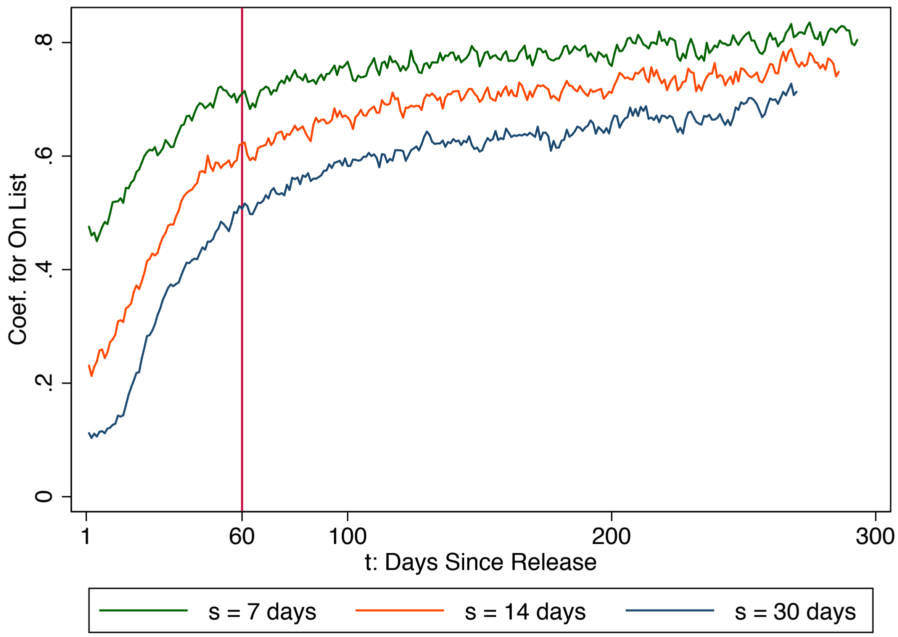
(b) Ranking



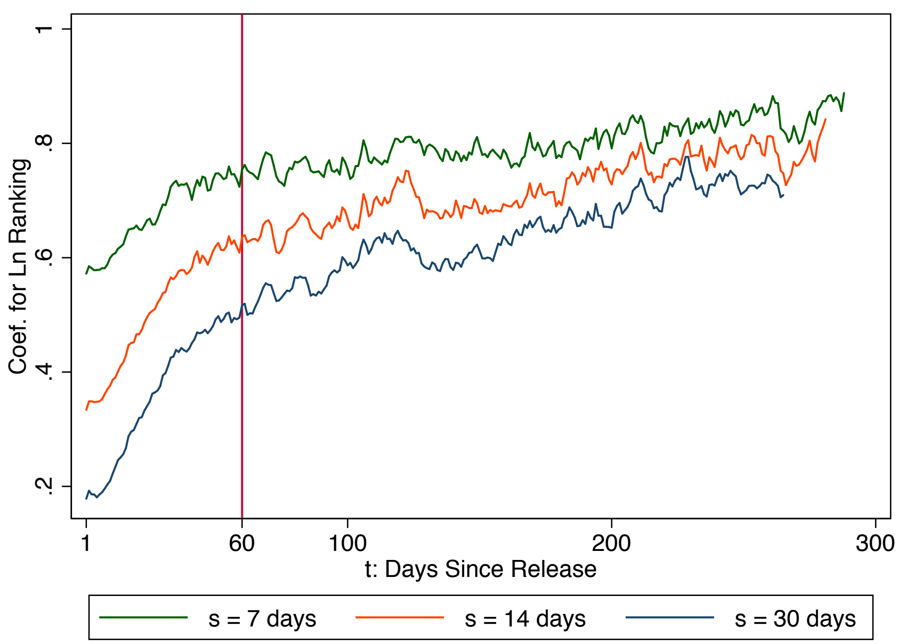
Notes: This figure plots the probability of on top list and average ranking across apps in four categories: apps on/off top list on day 12/100. Numbers in parentheses are counts of apps in that category.

Figure 8: Increasing Predictability

(a) On List



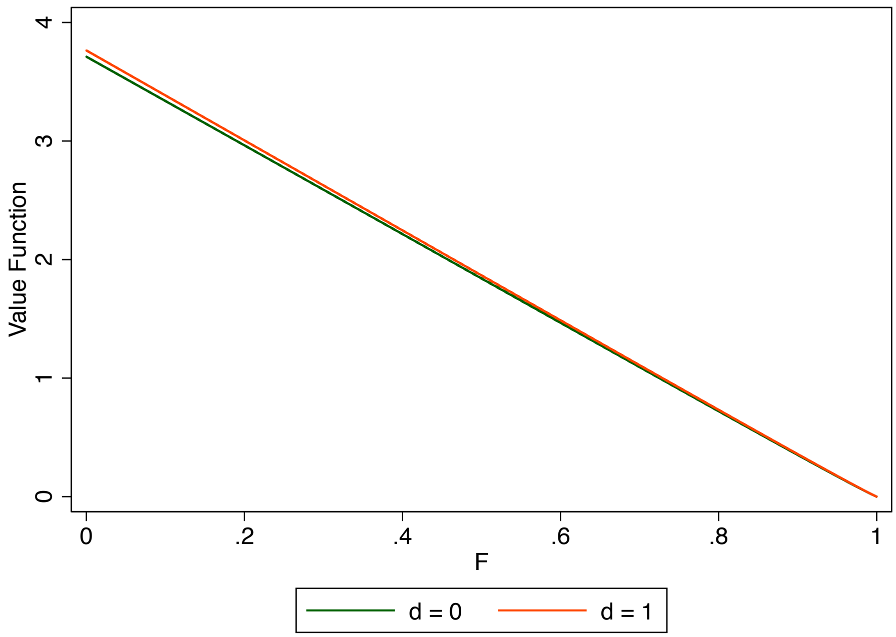
(b) Ln Ranking



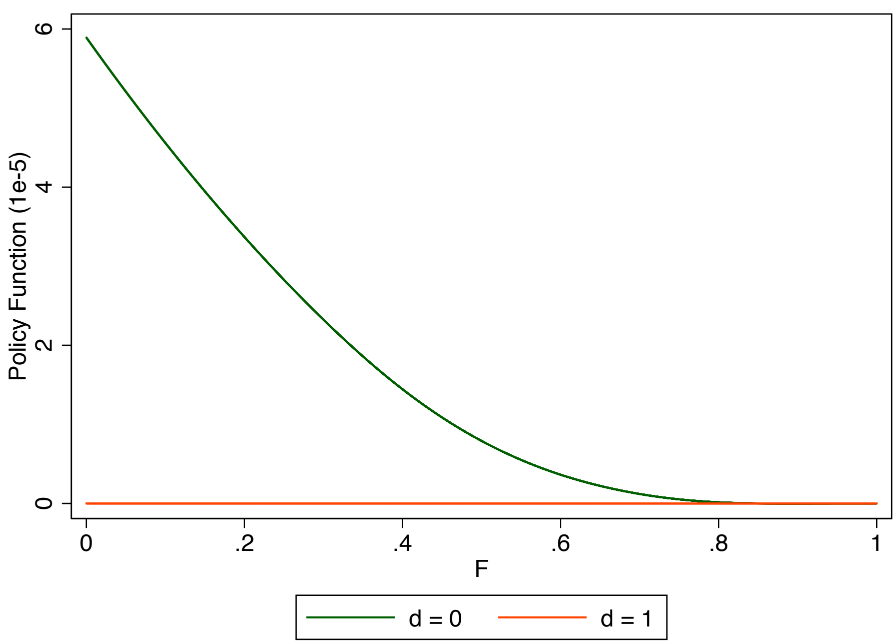
Notes: This figure plots the coefficients $\rho_{t,s}$ from regression $y_{i,t+s} = \rho_0 + \rho_{ts} \cdot y_{it} + \varepsilon_{it}$. y is dummy for top list on top panel, and logarithm of ranking on bottom panel. The x-axis measures the value of t, and different lines represent different values of s.

Figure 9: Simulation of the Modified Bass Model

(a) Value Function

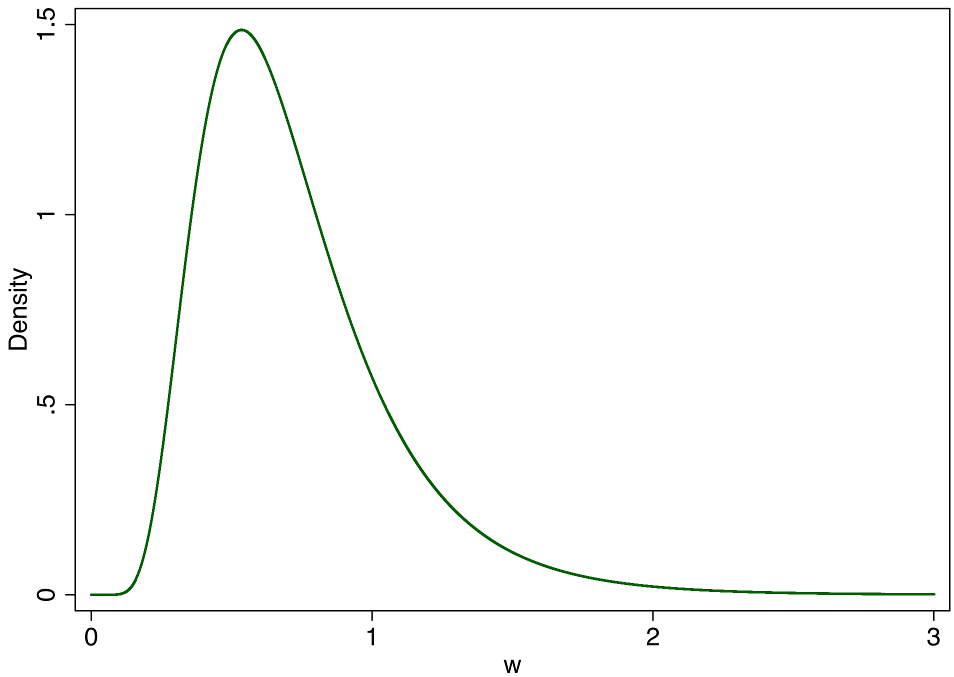


(b) Policy Function



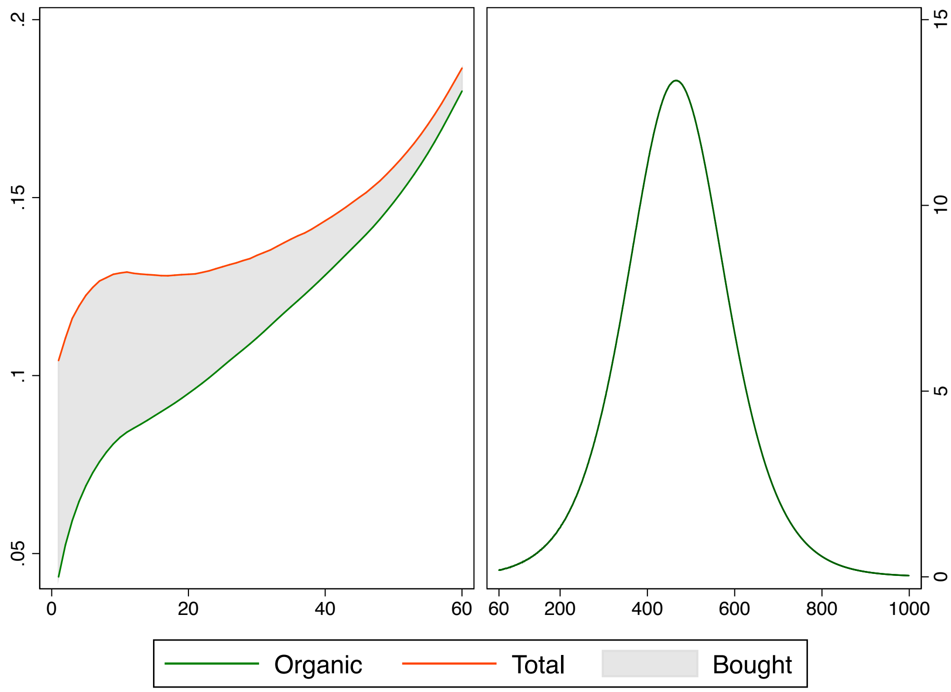
Notes: Simulation result of the model, with coefficient values as follows: $p = 0.149$, $q = 1.86 \times 10^{-05}$, $M = 4.07$, $\pi = 1$, $C = 2.25$, $\beta_{0R} = 0$ ($\beta_0 = \log(500) - \beta_{0R}$ to account for the sign change and censoring), $\beta = 0.23$ ($\beta = -\beta_R$ to account for the sign change), $\sigma_u = 1.61$, $\Phi = \ln(50)$, $\delta = 0.9999$.

Figure 10: Distribution of w



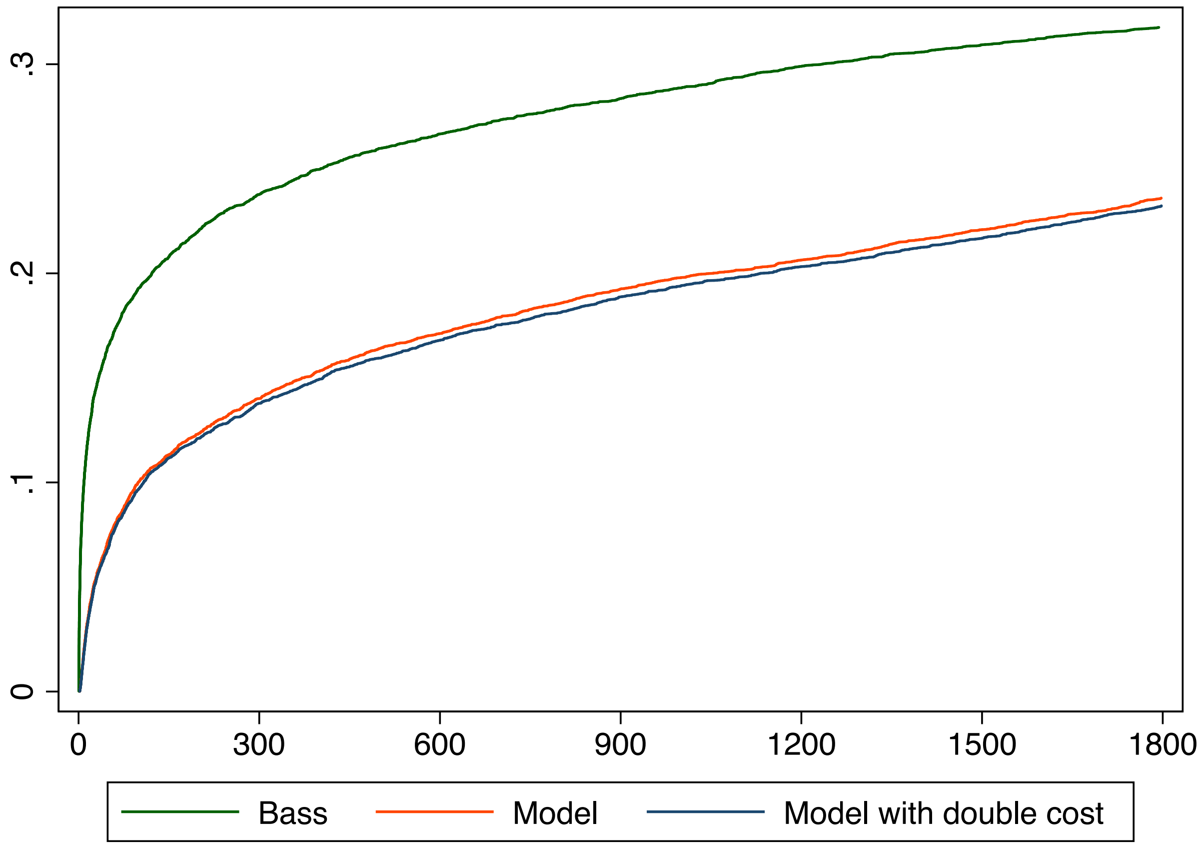
Notes: This figure plots the PDF of developers' willingness to pay for one organic download (as a proportion of the cost of a bought download based on the estimates reported in Table 5.

Figure 11: Diffusion of the Median App



Notes: This figure – it is one figure –plots the diffusion pattern of organic, bought, and total downloads for the median app reported in Table 6 under the assumption the app developer's BDL behavior follows our model. The X-axis represents days since release, and Y-axis represents the simulated volume of downloads. We have rescaled the Y-axis for X>60 (days) by 7.5x. The magnification of the early days permits seeing the period in which the developer is most active more clearly.

Figure 12: CDF for Half-life



Notes: These figures plot the cumulative distribution function (CDF) for half-lives (the day when half of the whole diffusion process takes place) under three scenarios: the standard Bass case, the current model, the current model with half the value per organic download. In each scenario, 10000 apps are simulated using parameters reported in Table 5.