Merger Analysis in the App Economy: An Empirical Model of Ad-Sponsored Media *

Kohei Kawaguchi† Toshifumi Kuroda‡ Susumu Sato§

July 14, 2020

Abstract

This paper proposes a new model of imperfect competition of ad-sponsored media for a merger analysis applicable to the mobile app industry. Our framework addresses problems inseparably linked to this industry. First, to catch up with newly created and quickly redefined markets, we automate the conversion from in-text product descriptions to numerical product attributes by combining word embedding and dimension reduction techniques. Second, to analyze developers' monetizing with both price and sponsored advertising in an app, we consider a consumer who faces both budget and time constraints. The model defines an equilibrium over consumers' downloads, usage, and in-app purchase decisions and app developers' price and non-price competition. We prove that an iterative algorithm converges to the least equilibrium. We estimate the model using mobile app data from Japan between 2015 and 2017. Based on the estimated model, we show that relevant market definitions that ignore either the download price or sponsored ads are misleading a merger analysis. A hypothetical split simulation of a major communication app shows that the total surplus can increase by 2.5% for the spin-off app. A merger simulation of the top apps in each category shows that "killing" acquired apps often maximizes the total profits of the merged company. Reducing the platform fee to zero can push up the total surplus by 8.4%.

Keywords: Merger simulation, relevant market, SSNIP, antitrust policy, ad-sponsored media, attention economy, app economy, distributed word representation, rigorous post-lasso.

JEL Codes: L11, L13, L41, L86, M13, M21.

*This study is financially supported by JSPS grant 18K12768 and Japan Broadcasting Corporation. At the time of the research, Kuroda was a member of NHK's Internet business committee and JFTC's Study Group on Improvement of Trading Environment surrounding Digital Platforms. The views and opinions in this paper are those of the authors and do not necessarily represent those of NHK and JFTC. We thank Akira Mizukami, Pedro Pereira, Masayuki Asahara and Taku Masuda for their helpful comments. We are also grateful for helpful comments and discussions at seminars at Kansai University, Kyoto University, International Christian University, ITS European Conference 2019, and SWET2019. We thank Akira Matsushita for the research assistance. All errors are our own.

†kkawaguchi@ust.hk, Department of Economics, School of Business and Management, Hong Kong University of Science and Technology

‡kuroda@tku.ac.jp, Department of Economics, Tokyo Keizai University

§susumusato@live.jp, Institute of Economic Research, Hitotsubashi University

1 Introduction

Defining a relevant market and conducting a merger simulation are cornerstone activities of an antitrust policy. Nonetheless, doing so is not straightforward in the app economy, which is playing an increasingly vital role in shaping the ecosystem of software platforms such as smartphones, tablets, and laptops. This difficulty occurs because markets are newly created and quickly redefined, and the co-existence of multiple monetizing policies such as *freemiums* prevents us from identifying substitution patterns based on a traditional method that primarily uses price variations. Therefore, the authority faces greater challenges when imposing conditions on deals. Thus, the uncertainty in the definition of a relevant market and the assessment using a merger simulation gives interest groups the possibility of manipulating antitrust policies. For example, when *Facebook* attempted to merge with *WhatsApp*, European telecommunication companies encouraged the European Commission to challenge the case because the merged entity would hold a dominant position in the "instant messaging" market (The Wall Street Journal, 2014). Whether this definition is too narrow or wide and how changes in market power are evaluated critically influence the decisions that the antitrust authorities make and their consequences. Any other ad-sponsored media such as newspaper and cable TV, poses the same problem regarding antitrust policies; product characteristics are difficult to describe, and multiple monetizing policies co-exist.

In this paper, we propose a new framework to define a relevant market, conduct a merger simulation of ad-sponsored media, and apply it to the mobile app industry. This framework introduces several new features to address the aforementioned problems. First, we use word embeddings to a semantic space (Deerwester et al., 1990) to convert a product description into a numerical vector that is used as a product characteristics vector in the consumer choice model. This approach allows us to catch up with a fast-growing market by automating the translation from product descriptions to numerical product characteristics. By referring to detailed information on product characteristics, we can also avoid colloquialisms when defining a relevant market. We integrate the resulting numeric representation of product descriptions into a consumer choice model and let the choice data reveal the substitution pattern across products. We apply a rigorous post-LASSO method (Belloni and Chernozhukov, 2013) by assuming sparsity in the manner in which the semantic vectors affect consumer choice to identify key dimensions in the product characteristics space. This entire procedure allows us to define a market supervised by choice data rather than based on an unsupervised classification solely using product descriptions.

Second, we consider a consumer who faces both budget and time constraints and explicitly model the time cost for a consumer to use a free service when watching advertisements. In this setting, in addition to the traditional pecuniary prices to download and use the apps, mobile app developers can effectively set the "price" by increasing mobile advertising intensity. Such an increase will raise the time cost for consumers and the revenues that a developer receives from advertisers. We explicitly model this non-price competition of mobile app developers regarding advertising intensity. Developers endogenously choose whether to charge download prices or display advertisements. Modeling this type of non-price competition of ad-sponsored media is particularly important to analyze the *attention economy* (Brynjolfsson and Oh, 2012; Bordalo et al., 2015; Allcott et al., 2020), an economy in which firms compete to attract consumers' time and impression.

One of the empirical problems is that we do not observe an app's advertising intensity. At best, we can only observe whether advertisements are shown in an app. Our approach is to elicit the advertising intensity set by developers by exploiting a unique feature of the mobile app industry: the direct marginal cost of acquiring sponsored advertisements is negligible. In other words, we use advertising optimality conditions to elicit unobserved advertising intensity instead of identifying the marginal cost parameter. We note that the marginal cost of increasing usage is identified from the pricing optimality condition as usual. The estimated model allows us to measure market power in not only monetary prices but also the new price, the advertising intensity and allows us to define a relevant market and conduct merger simulations with prevailing multiple monetizing policies. We estimate the model using detailed data about mobile download, revenue, and usage. We demonstrate that our model fits the data well in every aspect both in-sample and out-of-sample. We show that semantic vectors representing product information explain a significant part of the unobserved heterogeneity.

We conduct several policy experiments by relying on the estimated model for mobile apps. First, we apply the *Small, Non-transitory but Significant Increase in Price (SSNIP)* test to define relevant markets in the mobile apps market. Specifically, we demonstrate the biases in the relevant market definition when we ignore some "prices" such as advertising intensity. For instance, for a news app, for instance, compared with the SSNIP test with both price and advertising, the SSNIP test with only price finds relevant markets that are too small, whereas the SSNIP test with only advertising finds relevant markets that are too large. We also show that the order of the apps to check the change in profits attributable to SSNIP is a crucial issue. We show that a greedy strategy, which sequentially adds an app that maximizes the profit change using SSNIP from the remaining apps, works better than another way of orderingthe apps. The relevant market defined by the SSNIP using a greedy strategy is always smaller than the market defined by the product category specified in Google Play. Nevertheless, the market's share and the concentration are not necessarily higher in the relevant market defined by using the SSNIP test. The results indicate that an arbitrary definition of the relevant market can mislead an antitrust authority.

Our model enables us to conduct a full-equilibrium merger simulation. Unfortunately, we cannot name apps because of the confidentiality contract. We first conduct a split simulation of a major communication app that was once acquired by a company that owns another major communication app. The simulation results show that the total surplus increases by 2.5% from a spin-off app, by 0.1% from the parent company app, and by 0.6% from outsider apps. This split increases the consumer surplus at every app, and only hurts the parent company app's profits and the platform's revenue, Google Play. We next consider a hypothetical merger in which the top app in each product category acquires the second to fifth closest apps in terms of cross-elasticity. The magnitude of the effects differ across apps: however, the download price increases the most among the acquired apps, whereas advertisements increase the most among the acquiring app. As a result, both the consumer and the total surpluses drop from 0.5 to a few percents for each app. One of the interesting patterns that emerges from this analysis is that an increase in the price of acquired apps often results in these app's experiencing reduced profits. Thus, the acquiring developer finds that "killing" acquired apps that are close substitutes of the top app is a profitable post-merger strategy..

Finally, we conducted a counterfactual analysis of reducing the platform fee from 30% to zero. Such a reduction results in a moderate decrease in the download price and a substantial decrease in the sponsored advertisements. Two mechanisms worked behind this change. First, the price declines because the double marginalization is removed. Second, the price increases and advertisement decreases because download revenue is no longer taxed by the platform. The result is a 51.5% increase in the profits of the app and a 10.6% increase in the consumer surplus. In total, the surplus increases by 8.4%.

The remainder of this paper is organized as follows. In the rest of this introduction, we clarifies the novelty and contributions of our paper in combination with an overview of the relevant literature. Section 2 provides an overview of the mobile app market and its institution, and Section 3 explains the data we use for the analysis. Section 4 describes how to numerically represent the in-text app description. Section 5 lays out the model and proposes an algorithm to solve the model. Section 7 derives an estimator for the key structural parameters in the model. Section 8 defines the relevant market for several apps based on the estimated model. Section 9 conducts hypothetical split and merger simulations and evaluate the competitiveness of the mobile app market. Section 10 analyzes the effect of platform fee reduction, and Section 11 concludes the paper by restating the contributions and clarifying the limitations of the analysis.

1.1 Novelty and Contributions

Our structural model of competition among mobile app developers can be classified as a model of competition among ad-sponsored media (Anderson and Gabszewicz, 2006). This body of literature analyzes mergers among ad-sponsored media in an environment in which consumers are single-home, and advertisers are multi-home (Anderson and Peitz, 2020), or both consumers and advertisers are multi-home (Anderson et al., 2019). Our model belongs to the former framework. Our model differs from existing theoretical models of mergers among ad-sponsored media in one way: business models (paid media or free media) can change after a merger, whereas existing studies assume that the business model is exogenously given.

Some studies also analyzed in-app purchases as versioning strategies of a monopolistic two-sided platform (Jeon et al., 2016; Lin, 2020). Compared with these studies, by developing a simple model of consumers' in-app purchasing decisions, our model incorporates in-app purchases in an oligopoly framework. Some studies also analyzed the endogenous choice of business models as a device for strategic differentiation (Calvano and Polo, 2019) or as a form of second-degree price discrimination (Sato, 2019), among others, in a different environment. Our model uses non-negativity constraints for prices and advertising intensities to derive the endogenous choice of a business model: when the non-negativity constraint for download price binds, the app is provided for free, and when the non-negativity constraint for advertising intensity binds, the app is provided without advertisements. Given the heterogeneity in app and developer features, this characterization enables an analysis of the co-existence of multiple business models in a single framework.

Some studies used text data for an economic analysis. Each such study numerically represented different information in text data in various ways. Gentzkow et al. (2019) reviewed the exploding body of literature of various fields of economics research using text as data. In the mobile app industry, Liu (2017) and Ershov (2020) used app descriptions to categorize apps. Deng et al. (2018) used app's descriptions to study differences in functions between their paid and free versions. Leyden (2018) used the descriptions of app's release notes to define product categories and distinguish bug fixes and feature updates. Pervin et al. (2019) evaluated user reviews as positive, negative, and neutral. Barlow et al. (2019) and Angus (2019) used product descriptions to measure the similarity of apps. Existing studies manually processed text data, counted word frequency, or used sentiment analysis. Our study differs from their method by using product characteristics represented by a semantic vector obtained through word embedding (Deerwester et al., 1990; Mikolov et al., 2013b).

The following papers used information elicited from text data as part of the product characteristics in demand estimation. Gentzkow and Shapiro (2010) used a slant measure based on text data to estimate the demand for newspapers. Ghose and Han (2014) and Kesler et al. (2017) used several pieces of information in product descriptions such as file size, version, and number of characters as product characteristics. Kwark and Pavlou (2019) judged whether a good is a substitute or a complement for other goods based on product descriptions and then studied the effect of a product's consumer review on its substitutes and complements. Leyden (2018) used the aforementioned information to estimate demand. Following the approach in Ackerberg and Rysman (2005), Ershov (2020) used the number of products in the categories to control for unobserved product characteristics approach. Ours is the first paper that uses high-dimensional embedding for words in product descriptions as product characteristics to estimate consumer demand.

The body of literature on mobile app demand estimation is growing. Carare (2012) and Ifrach and Johari (2014) estimated the effect of mobile app store rankings on demand. Ghose and Han (2014) estimated the discrete choice random coefficients demand for mobile apps that considers various product characteristics, including in-app purchases, in-app advertising, and the number of updates as fixed characters. Han et al. (2016) estimated a consumer choice model on both mobile app downloads and usage through a discrete-continuous choice framework. Ershov (2020) examined consumer product discovery costs for game apps on the Google Play platform. Leyden (2018) estimated the dynamic discrete choice of a consumer over mobile apps to investigates the effect of product updates. Our paper differs from these studies in multiple dimensions. First, we consider both download and usage decisions over mobile apps. The only exception is Han et al. (2016). However, their data and analysis are at the product category level, whereas ours is at the product level. Second, we explicitly model the interaction between advertising intensity and consumer download and usage choice. Ghose and Han (2014) and Leyden (2018) included an advertisement dummy to estimte demand, but did not consider advertising intensity. Third, we include high-dimensional product characteristics elicited from in-text product information, allowing us to avoid an assumption about the product category to which each app belong. Thus, we do not restrict the substitution pattern based on a pre-specified product category. Finally, our data cover a wider variety of mobile apps.

Several papers studied the strategy of mobile app developers. Ghose and Han (2014) considered the price competition faced by mobile app firms. Ershov (2020) investigated the pricing and entry as a firm strategy. Leyden (2018) investigated the pricing and update strategy of mobile apps. Liu (2017) investigated app developers' choice of platform. Our paper differs from these studies by jointly considering the pricing and advertising strategies. Our paper is the first to explicitly model and empirically analyze the imperfect competition of mobile app developers over consumer app choice and time usage.

Some studies included the opportunity cost of time usage in a consumer decision problem. Jara-Díaz and Rosales-Salas (2017) reviewed time use studies in transportation research that range from purely descriptive studies to econometric modeling analyses. Regarding time usage in the digital economy, Goolsbee and Klenow (2006), Brynjolfsson and Oh (2012) and Pantea and Martens (2016) used the opportunity cost of usage time to evaluate the value of free digital services on the Internet. Han et al. (2016) estimated the utility and satiation of mobile app usage with a multiple discrete-continuous choice model at the app category level. Regarding competition among ad-sponsored media, Crawford et al. (2018) studied households' time allocation problem over TV channels to investigate vertical integration in the TV market. The novelty of our paper is that it integrates into the analysis the supply side's response in advertising. In our model, mobile app developers compete over the time spent by consumers, which affects how consumers allocate time across activities by strategically setting in-app advertising intensity.

Competition authorities in developed countries are concerned with potential anti-competitive practices in the digital economy. However, differences exist in the status of merger regulations. The Japan Fair Trade Commission (JFTC) addressed non-price competition by revising in December 17, 2019, its merger guidelines (Japan Fair Trade Commission, 2019) to evaluate the competitive impact of a merger on the characteristics of content, qualities, and user-friendliness when defining product and geographic ranges in digital services. Nevertheless, Crémer et al. (2019) pointed out the practical difficulty in obtaining a precise measure of digital service quality. The U.S. Department of Justice set up a task force to monitor the information technology industry that addressed these issues. The literature has provided several approaches to defining the relevant market for a product offered through ad-sponsored media. Emch and Thompson (2006) proposed using the sum of the prices of both sides to conduct a version of a hypothetical monopolist test in payment card networks. Evans and Noel (2008) proposed using a relevant market definition based on a critical loss analysis of multi-sided platform. They applied the concept to Google's acquisition of DoubleClick. Filistrucchi et al. (2012) investigated mergers of newspapers using a two-sided market model. Affeldt et al. (2013) extended the concept of the Upward Pricing Presser to two-sided markets and applied it to a hypothetical merger in the Dutch daily newspaper market.

Our paper is the first to provide a framework for relevant market definitions when a product's retail price can be free in an equilibrium. Our model differs from the literature on two-sided markets in that either the retail price or advertising can be at the zero boundary or in the interior. In other words, app developers can endogenously select different monetization modes, including zero prices with advertisements, positive prices without advertisements, and both.

Regarding the relevant market definition of mobile apps, previous papers regarded the product category as a relevant market. Ghose and Han (2014) and Ershov (2020) used product categories to set up a nested-logit model. Liu (2017) and Leyden (2018) focused on a few categories of apps, namely, game and productivity apps. We elicited the relevant market for mobile apps from the top apps in Google Play by developing a new framework for estimating the demand for mobile apps.

Certain papers conducted merger simulations among ad-sponsored media. Some of them studied the newspaper industry (Filistrucchi et al., 2012; Fan, 2013; Gentzkow et al., 2014; Van Cayseele and Vanormelingen, 2019). Others studied the radio (Jeziorski, 2014) and magazine (Song, 2011) industries. Our paper is the first to simulate a horizontal merger, in which suppliers can choose monetization mode over retail prices and advertising, and different monetization modes co-exist in the market. Previous papers identified the marginal costs of printing, producing, and acquiring new advertisements from advertising optimality conditions. These costs do not exist in the app economy. App developers can use a Software Development Kit (SDK) for an ad network to automate advertising. We exploit this unique feature of the app economy to elicit advertising intensity through the condition of advertising optimality.

Apart from an antitrust policy, the welfare effect of digital services has also been investigated. Goolsbee and Klenow (2006) and Brynjolfsson and Oh (2012) evaluated the economic value of free digital services using the opportunity cost of time. Brynjolfsson et al. (2019) and Allcott et al. (2020) estimated the willingness to accept digital services by conducting choice experiments. We differ from these studies by estimating the demand function to evaluate the welfare effect of consumer surplus.

Our paper is also related to the literature on the welfare effects of new products. Hausman (1996) used a product space approach with an almost-ideal demand system and the constant elasticity of substitution utility to evaluate the welfare effect of new products. More recently, the welfare effects of new products are evaluated using a product characteristics approach following (Berry et al., 1995), such as for minivans (Petrin, 2002), personal computers (Eizenberg, 2014), and mobile apps (Ghose and Han, 2014). Berry and Pakes (2007) and Song (2007) used pure characteristics models to avoid a mechanical increase in welfare from new products. Morozov (2019) found that the limited adoption of new products can be mostly attributed to search frictions by estimating a demand model with consumer searches in the U.S. hard drive market. Our paper integrates high-dimensional product characteristics elicited from in-text product information and considers the discrete-continuous choice over download and usage to evaluate the welfare effect of new mobile apps. Our model includes idiosyncratic choice-specific shocks: that is, it is not a pure characteristics model.

2 Industry Background

2.1 Mobile App Industry

Although complete information on the global app economy is unavailable, several reports provide a fragmented view of this rapidly growing app economy. We sketch the landscape of the app economy during the data period from 2015 to 2017.

Mobile app Mobile app is application software designed for mobile devices, such as smartphones and tablets. Smartphones and tablets are multi-purpose mobile computing devices that typically have a touchscreen, Internet access, camera, microphone, speaker, and a specific operating system (OS) that manages the hardware and software. The distinction between smartphones and tablets is unclear. However, smartphones usually provide mobile data access through a cellular network and are smaller than seven inches.

As of 2017, *Android* and *iOS* are the two mainstream OSs. Android is developed by *Google* and iOS by *Apple*. In 2017, the OS market share in smartphones was 73.5% for Android and 19.9% for iOS.[1] The market share in tablets was 29.0% for Android and 70.7% for iOS.[2]

Mobile apps take up a significant amount of Internet usage time. App Annie (2017) reported a breakdown of the time spent using the mobile Internet in selected countries. The report indicated that consumers in both developed and developing countries spent more time on mobile apps than on mobile web browser. For example, in the United States, the ratio of app usage time is 88%. Moreover, consumers are increasingly using the Internet through the mobile Internet. comScore (2017) showed that the 2017 mobile share in the United States was 65%. Thus, understanding consumer behavior in the mobile app industry is essential for understanding consumer behavior on the Internet.

Mobile app stores Consumers can download and install mobile apps from online stores for both OSs. Some apps are free to download, and others have a price attached to them. Mobile apps for iOS can be downloaded only from the *Apple App Store* but can be downloaded from several stores for the Android OS. Google operates the Google Play mobile store, and other mobile app stores for Android include *Galaxy Store* for *Samsung* devices and *Epic Games*. Nevertheless, in 2017, the majority of the downloaded apps were still from Google Play and the Apple App Store. To distribute a mobile app through a mobile app store, the developer has to pass a review process, and these processeshave review policies that differ across mobile app stores. [34]

Mobile app stores classify apps into several categories, such as games, music, and news, to

[1]Mobile operating system market share worldwide in 2017, statcounter, `http://gs.statcounter.com/os-market-share/mobile/worldwide/2017`

[2]Tablet operating system market share worldwide in 2017, statcounter, `http://gs.statcounter.com/os-market-share/tablet/worldwide/2017`

[3]The review guideline for App Store `https://developer.apple.com/app-store/review/guidelines/`.

[4]The guideline for Google Play `https://play.google.com/intl/ja/about/developer-content-policy-print/`
.

enable consumers to easily find a desired app. A mobile app has a page on each mobile store that provides information about the app. Figure 1 provides an example from 2018 of a page on Google Play.

Mobile app developers Mobile app developers face two-sided markets: consumer and advertiser. Developers can earn revenues from both sides, and both sides of the market grew rapidly during the data period. According to App Annie (2017), the number of mobile app downloads increased by 60% between 2015 and 2017 and amounted to more than USD 175 billion in 2017. App Annie (2019) also reports that mobile ad sales increased by 30% during 2017 and mobile ads were expected to account for 62% of global digital ad spend in 2018, representing USD 155 billion, an increase from 50% in 2017.

Revenues from consumers consist of priced downloads and in-app purchases. The download price is usually charged only when a consumer downloads the app for the first time. A consumer who purchased an app is allowed to download the app multiple times without paying extra and can use the app on multiple devices. Of course, some apps restrict the number of devices on which a consumer can use them or issue licenses that restrict this number.

An app developer can also collect in-app purchases through mobile app stores. A consumer pays within an app to remove restrictions on the app's functionality or to upgrade the service. For example, a consumer may pay to suppress mobile ads or purchase an item in a game. If consumers pay the download price or make in-app purchases through mobile app stores, these stores charge the app developers a transaction fee. For example, Google Play and the Apple App Store charge a transaction fee of 30%. As a result, the app developer can earn only 70% of the app price and in-app purchases. Some app developers attempt to charge for apps outside mobile app stores: however, Google Play and the App Store forbid this practice through their review guidelines. For example, an update to the *Spotify* app was rejected in 2016 because the app attempted to lead consumers to use an outside payment platform. Spotify claimed that the rule was used to protect *Apple Musice* (Recode, 2016). In 2016 and 2018, Apple and Google, respectively, reduced transaction fees to 15% for consumers whose subscription terms went beyond one year[5][6].

Mobile ad networks Another source of revenue for a mobile app developer is advertising fees that advertisers pay to display their advertisements on the app. Most advertisers and mobile apps use a service that connects advertisers and websites or apps, an *ad network*, to distribute and host advertisements. Some mobile apps choose not to use an ad network and sell advertising space directly to advertisers. They do so to reduce the transaction fees paid to ad networks and to target specific advertisers by taking advantage of their app's unique customer base.

In 2018, more than 250 mobile ad networks were in operation.[7] An ad network distributes

[5]The announcement from Google is here. `https://support.google.com/googleplay/android-developer/answer/112622?hl=en`

[6]The news reports Apple's reduction of transaction fee is here.https://techcrunch.com/2016/06/08/apple-to-introduce-search-ads-on-app-store-along-with-changes-to-app-review-discovery-and-splits/

[7]`https://www.appsflyer.com/2018indexpage/`

software development kits (SDK) to integrate ads into mobile apps. An ad network then allows advertisers to specify parameters, such as region, device, OS, interests, and gender, to determine the target audience. Advertising space is usually transacted through an auction. For example, in Google's *AdMob* ad network, advertisers can bid on a per click or impression basis. AdMob ranks between click bids and impression bids in order of expected revenue to predict the likelihood that a click bid ad will be clicked. For the developer side, mobile app developers set a price floor. Then, AdMob distributes ads only to websites and apps that have expected revenue higher than the price floor. AdMob also provides advertisers with an optimizer that dynamically sets price floors depending on a geographic location, traffic, and other pieces of historical data.[8] Other than AdMob, other services assist mobile app developers with hosting mobile ads through multiple ad networks. *InMobi* provides an ad mediation platform that assists mobile apps with hosting mobile ads from the highest bidder across multiple ad networks.[9] Because of the high number of ad networks and apps that accept ads in the market, the cost of ad-network is far lower than direct selling.

Recent antitrust and merger cases During the past two decades, tech giants including Facebook and Google acquired many start-up firms. Google acquired *YouTube* for USD 1.65 billion in 2006. App Annie (2017) reported that YouTube was the most used video streaming app in the United States in 2017. This acquisition completed this case in early termination. In contrast, Facebook acquired Instagram in 2012 for USD 1 billion and WhatsApp in 2014 for USD 19 billion. The antitrust authority in the United States and the European Union approved these mergers after a detailed merger review. App Annie (2017) reported that Facebook, Messenger, and Instagram are the top three apps by monthly active users in the United States. In addition, WhatsApp is the most used social app in Germany, Indonesia, India, Russia, Spain, and the United Kingdom, and its merger of Facebook appears to have relaxed the market competition in the social app market. In addition, the European Union fined Facebook EUR 110 million (USD 122 million) for providing misleading information on its merger with WhatsApp. However, whether the European Union could have blocked Facebook's acquisition of WhatsApp using horizontal merger regulations if it had the correct information is unclear. Furthermore, whether Facebook and WhatsApp compete in the social app market is unclear. In addition, the social app market may be too narrow to define. An assessment of whether social apps are a good subset of apps should made to understand the competitive environment in which a firm operates.

3 Data

The data we use to estimate the model come from several sources. First, we use the data provided by the consulting company *App Annie* to construct app download, usage, in-app purchase, and market size data. Second, we collect information on Google Play using the web scraping method and combine them with similar data provided by App Annie to complete the product description and

[8] `https://support.google.com/admob/answer/3418058?hl=en`
[9] `https://japan.inmobi.com/advertising-cloud/mediation`

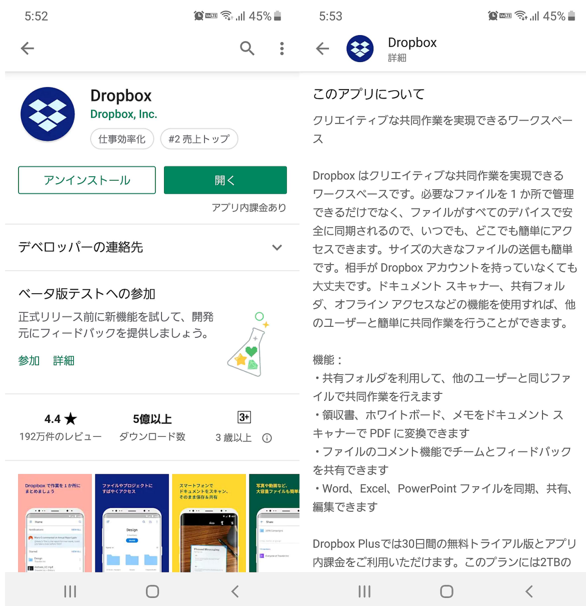


Figure 1: Product description in Google Play

characteristics data. Third, we use data provided by the mobile ad platform *Adtapsy* to construct unit advertisement price data. Because the App Annie database contains complete information on iOS only after June 2018, when we lack information on advertisement price data, the subsequent analysis focuses on Android apps.

3.1 Download, Usage, Download Price and In-app Purchase

Source App Annie is a consulting company that surveys, collects, assembles, processes, and sells a mobile app database. The App Annie API allows us to extract data on a wide variety of apps in more than 150 countries worldwide that are distributed through the App Store or Google Play. The company combines statistical models and procedures to estimate download, usage, revenue, and several other variables of each mobile app using data from key mobile app stores, key ad networks, proprietary consumer panel surveys, in-app tracking information, and publicly available data. App usage is defined as the number of minutes that it runs in the foreground. Apps in the background are not recorded as in use.

Coverage, period, and selection Because the same app can be sold with different names across different platforms, the company assigns a unique identifier to each app. We use the list of unique app identifiers as the list of products. The company classifies apps first into "Game" and "Application" and then into finer categories, such as news, music, and education apps. For every unit of the observation period daily, weekly, and monthly the company calculates for each app the number of downloads, the revenue, and its rank within each category. The API only allows us to access data on the top 1,000 apps in each sub-category and during a period for each variable. We use daily data as the baseline, if available, and aggregate them depending on the type of analyses. For variables that are only available weekly, we use weekly data. Because the day $\times$ category is a fine enough segment, apps below the top 1,000 have almost zero downloads and revenues.

The data are available since March 2010 for iOS apps and since January 2012 for Android apps. However, because the unit advertisement price data are only available from March 2015, we use data between March 2015 and January 2017 in the estimation. We select the set of apps to be analyzed in the following manner. First, we use information on price and whether each app appears in-app advertisements to classify apps into three business models: free advertising apps, paid advertising apps, and paid non-advertising apps. Next, we compute the fraction of each business model relative to free advertising apps. Using this fraction, for each week and business model, we select the apps ranked higher than the threshold rank, defined by 100 multiplied by the fraction of the business model relative to free advertising apps. The ranking is in either usage time or number of downloads. Finally, we select the apps ranked higher than the threshold rank of 10 times or more in usage or download. The selected apps are the set of apps to be included in the sample.

For each app selected busing these criteria, for some weeks, download, usage, or revenue information is missing because the app was not ranked higher than 1,000 but may have operated during those weeks. We fill in these missing values by substituting the minimum value of the observed data in the same categories.

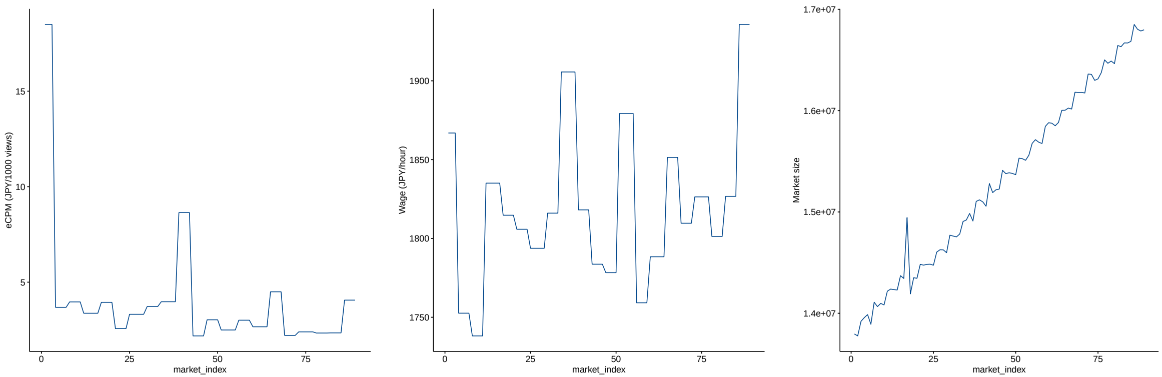
Variables For each app, the data contain product name, developer name, parent company name, product category, devices available, release date, and download price if it is not free. For each app and period (daily, weekly, and monthly), the data contain the number of active users, which is defined by the number of unique users who opened the app during each period (only weekly), the usage penetration rate, which is defined as the number of active users divided by the number of active devices (only weekly), the number of downloads, the average time spent by active users (only weekly), revenues during the period, and several other variables that are not used in our analysis. The one drawback to the data is that the price information does not reflect sales discounts.

Because an app's revenues include the revenues from both downloads and in-app purchases, we subtract the price times the number of downloads from the revenues of each app to calculate the revenues from in-app purchases. The in-app purchase per user is calculated as the revenues from in-app purchases divided by the number of downloads.

We multiply the number of active users and the penetration rate of an app to calculate the estimated number of active devices. The numbers are supposed to coincide across apps but are slightly different because of App Annie's calculation process, we take the average across the apps. We use this value as a proxy for the size of the consumer base for mobile apps. We assume that a consumer has a unit download demand per day and define the market size as the number of active devices multiplied by the number of days in a period. To ensure that the sum of the market shares in each period does not exceed one, we multiply the estimated number of active devices in

Table 1: Summary statistics at the week/app-level

	N	Mean	SD	Median	Min	Max
Usage time (Hour/User)	41560	2.233	2.126	1.421	0.506	23.905
In-app charge per download (JPY)	41560	1159.810	4797.084	0.000	0.000	231610.579
Download	41560	20868.980	43918.720	3697.024	2.246	831602.250
Download price (JPY)	41560	245.606	479.718	0.000	0.000	3470.760



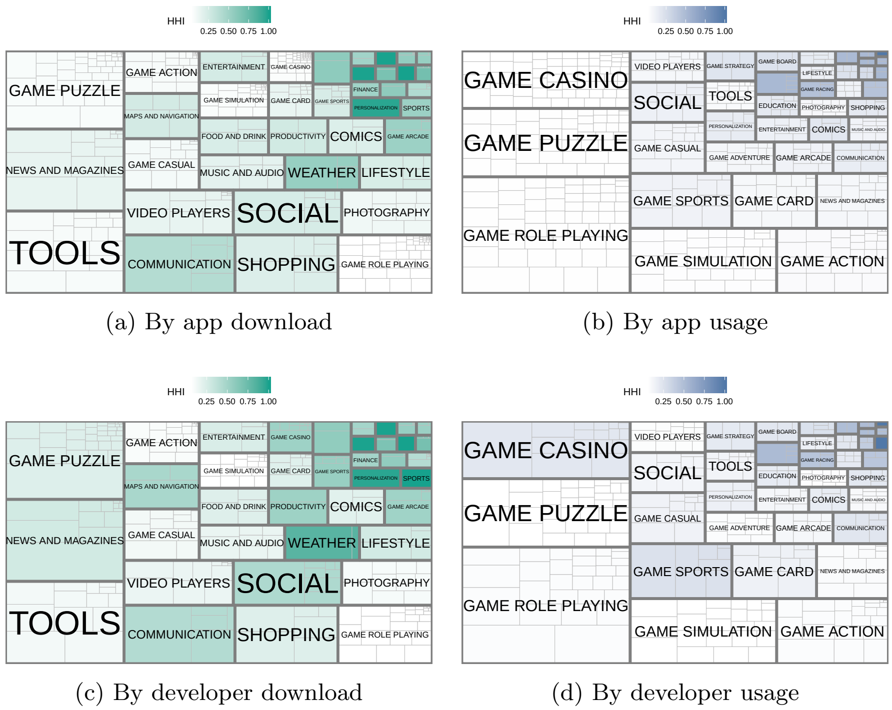
(a) Advertising price (b) Hourly wage (c) Market size

Figure 2: Summary statistics at market-level

all periods by a constant number.

Summary statistics Table 1 provides the summary statistics for usage time, in-app purchase per download, number of downloads, and download price. Figure 2 summarizes the time series for advertising price, hourly wage, and market size, and indicates a jump in market size during the week starting from September 3, 2015. Although the exact reason for this jump is unclear, the event that Google Play Music launched in Japan on September 3, 2015 might have affected the original data. [10] Figure 3 demonstrates the market share and Herfindahl-Hirschman Index (HHI) based on the Google Store categories. The upper panel measures the share by the number of downloads, and the lower panel measures shares by usage time. Figure 2 also indicates that market share based on downloads and usage can be substantially different. Specifically, games have a larger market share when measured by usage time than when measured by the number of downloads. Figure 4 illustrates the paid app and with-advertisements app shares in each app category, which reveals a negative correlation in the shares across categories. This negative correlation indicates that charging a download price and hosting advertisements are substitutable ways for app developers to monetize. Thus, pricing and advertising must be jointly analyzed.

[10] https://japan.googleblog.com/2015/09/google-play-music.html



(a) By app download

(b) By app usage

(c) By developer download

(d) By developer usage

Figure 3: Market share and Herfindahl-Hirschman Index (HHI) defined by Google Store category

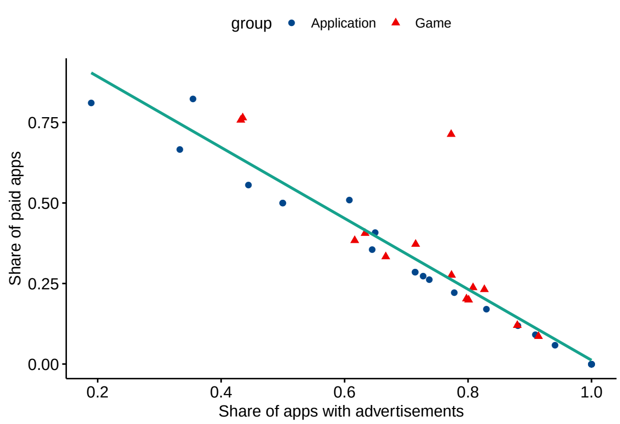


Figure 4: Share of apps with advertisements and download prices

3.2 Product Description

Source We use product descriptions displayed on Google Play to construct the advertising dummy, product class, and semantic vectors. App Annie records a history of product descriptions but the history of only one language's description for each app. Therefore, we mainly used our original data constructed by scraping the websites of Google Play as of December 2018. We generate numerical vectors of product characteristics from Japanese descriptions because these descriptions typically contain richer information, and consumers in Japan are likely to read Japanese text.

Advertising Dummy Our original data scraped from Google Play contain information that indicates whether or not the app shows ads. We define the advertising dummy as having a value of 1 when the app's store page contains "Contains Ads" strings in a predetermined place.

App Category We reclassify Google Play's 49 categories[11] into five product classes to avoid estimating a priori substitution patterns. One of our objectives is to define markets from a flexible demand substitution structure. For the logit demand specification, category dummy variables with random-coefficients provide nested-logit-like substitution patterns (Train, 2009) and Grigolon and Verboven (2014) found that a nested structure has a significant effect on the market definition.

For the parameter estimation, we reclassify the "Application" apps' category into the "Tools", "Information", and "Shopping" product classes and the "Game" apps' category into the "Hardcore Game" and "Casual Game" product class. The "Tools" product class contains the Art & Design, Beauty, Business, Communications, Dating, Libraries & Demo, Maps & Navigation, Personalization, Photography, Productivity, Social, and Tools categories. The "Information" product class contains the Books & Reference, Comics, Education, Food & Drink, Health & Fitness, Lifestyle, Medical, Music & Audio, News & Magazines, Parenting, Sports, Video Players & Editors, and Weather categories that are expected to obtain information. The "Shopping" product class contains the Auto & Vehicles, Entertainment, Events, Finance, House & Home, Shopping, and Travel & Local product classes that are expected to support commercial activity. Regarding Game apps, the "Casual Game" product class contains the Arcade, Board, Card, Casino, Casual, Educational, Trivia, and Word categories and the "Hardcore Game" product class contains the Action, Adventure, Music, Puzzle, Racing, Role Playing, Simulation, and Sports, and Strategy categories. Because app categories in Google Play are set by app developers, the rules for categoring apps are not uniform.

Morphological analysis To use product descriptions data in demand estimation, we first convert the sentences in the product description into a bag of words. In Japanese, because words are not separated in a sentence, we employ a morphological analysis engine to split Japanese sentences into

[11] For detail, refer to their website: `https://support.google.com/googleplay/android-developer/answer/113475?hl=en`.

a bag of words. Specifically, we use an open-source program called *MeCab* (version 0.996) (Kudo, 2005). MeCab is widely used in the Japanese natural language processing literature to decompose sentences into a bag of words. We use a neologism dictionary for MeCab as a word dictionary, which can be downloaded from the developer's website (Sato et al., 2017). This bag of words is converted into multidimensional vectors using the methods described in section 4.

3.3 Advertisement Price

Source *Adtapsy* is a mobile app advertising platform that matches advertisers with app developers and distributes advertisements through matched apps. Adtapsy is connected to several global ad networks that oeperates in Japan, such as AdMob, *AdColoy*, *InMobi*, *AppLovin*.

CPM and eCPM The advertisement price that an advertiser pays to show a unit of an advertisement on an app is determined through auctions and can differ across ad networks, advertisers, apps, and devices. A popular buying method is based on CPM, or Cost per mille (Latin word for thousands), and represents a fixed price to buy 1,000 ad impressions. If an advertiser buys an ad only in CPM unit, the actual price to buy 1,000 ad impressions coincides with the CPM. However, in reality, ad impressions are transacted in various units and formats. Therefore, *eCPM*, or the effective CPM, is the actual costs per 1,000 ad impressions, and is often used as a measure of the market price of ad impressions. According to Adtapsy's estimates, the average eCPM was USD 5.4 and USD 6.3 for Android and iOS in March 2015 and USD 2.2 and USD 2.9 in January 2017, respectively.

Market average eCPM Adtapsy has published a monthly time series of market average eCPM since April 2015. Because these data are the only ones available for mobile advertisement prices, to the best of our knowledge, we use them as the market price of an ad impression in the mobile app industry. The price is unbiased if the ad impression is a homogeneous product but can be biased to the degree that the mobile app is differentiated in the mobile ad market. Thus, our analysis should be extended with reservations for large mobile platforms with impressions when such platforms may have different values and may have market power in the mobile ad market.

3.4 Auxiliary Data

We use wage data as a proxy for the opportunity cost of mobile app usage. We obtain wage data for each age and gender class from the *Basic Survey on Wage Structure*, a survey by the *Labour Standards Inspection Offices*.[12]

[12] https://www.mhlw.go.jp/english/database/db-l/wage-structure.html

4 Numerical Representation of Product Description

We use *nwjc2vec*, developed by Asahara (2018), to numerically represent in-text product descriptions. *nwjc2vec* is publicly available for Japanese language embedding and employs *fastText* (Bojanowski et al., 2017) as the model and the *National Language Web Corpus* (Asahara et al., 2014) as data. Several implementations of *word embedding* methods exist, including *word2vec* (Mikolov et al., 2013b) and *GloVe* (Pennington et al., 2014). fastText is a popular implementation that incorporates *distributional statistics* and word-internal structures into word embeddings. We use nwjc2vec to transform app descriptions into 300-dimension semantic vectors.

4.1 Distributional Hypothesis and Word Embedding

The algorithm is based on the so-called *distributional hypothesis* (Firth, 1957). For example, consider a situation in which the weather news reports today's weather. Both "sunny" and "raining" fit into a context such as "It's ___ today". A word such as "birthday" could also fit into the context. However, we consider a sentence such as "It has been ___ lately", to which "sunny" and "raining" fit but "birthday" does not. In this way, we can construct a matrix that records the fit of words into different contexts. This matrix is called the distributional statistics of words, and a column in this matrix is called a corpus. The distributional hypothesis assumes that the similarity in the distributional statistics implies similarity in the semantics.

We can estimate lower-dimensional numerical vectors of real numbers that well approximate the meanings of words represented in distributional statistics. The resulting lower-dimensional semantic vector space is called *word embeddings*. Word embeddings are obtained by maximizing the likelihood of distributional statistics under a model that predicts a corpus. The model differs in how it defines contexts and relates contexts with words. The model in nwjc2vec uses local word neighborhoods in a sentence as contexts as well, as in Mikolov et al. (2013a) and Bojanowski et al. (2017). The model uses both *continuous bag-of-words (CBOW)* and *skipgrams* to relate contexts to a corpus. We use a model based on skipgrams, because they are often reported to outperform CBOW (Mikolov et al., 2013b; Eisenstein, 2019).

In addition to distributed statistics, fastText also exploits word-internal structures to estimate word embeddings. fastText assumes that a word vector should be consistent with the sum of the vectors of n-grams in the word. For example, the word "phone" is divided into n-grams such as "pho", "phon", "phone", "hone" and "on". The model assumes that two n-grams sharing either similar former or later n-grams also share a similar meaning. By doing so, the model predictions are robust to differences in tenses such as "go", "goes", and "gone"; forms such as "decide", "decision", and "decisive"; and synonyms such as "economy", "economic", and "economist".

4.2 NWJC2VEC

The National Language Web Corpus used by nwjc2vec is a Japanese language corpus constructed by the *National Institute for Japanese Language* that targets the 10 billion words used on web sites.

The corpus first used a program called *Heritrix* to crawl approximately 100 million URLs every three months starting in October 2012. The version of nwjc2vec that we use to build the product characteristics uses data crawled from October 2014 to December 2014.

Hyperparameters exist to train the fastText model. nwjc2vec chooses 300 as the dimension of word embeddings, a local neighborhood size of h_{max} 8, the number of negative samples of 25, and the range of character lengths of n-grams of 3 to 6.

We added product descriptions in Google Play and App Store to the training data and re-estimated word embeddings. However, because the word embeddings were almost unchanged, we use nwjc2vec's original word embeddings to evaluate product descriptions.

4.3 Conversion Procedure

We use nwjc2vec to convert app descriptions described in Section 3.2. We use an open-source *Python* library *gensim* (version 3.7.2) (Řehůřek and Sojka, 2010) to construct numerical representations of product descriptions as follows.

1. Build data that record each app's identifiers and bags of words of the app's descriptions.
2. Load nwjc2vec using gensim.models.faxttext.
3. Convert all words in the bag of words for each app into multidimensional vectors using gensim.models.fasttext.
4. Take an average of the multi-dimensional vectors of words in each app's description as the product characteristics of the app.

In the end, the number of converted words in the app descriptions is 41,789, which seems relatively smaller than the number of words in the product descriptions, at 186,553. This result depends on the fact that nwjc2vec was trained by a corpus up to December 2014. The store description was scraped in 2018 and we use a neologism dictionary developed in August 2018 for morphological analysis. Therefore, although nwjc2vec has 1,267,080 words in the training data, it does not record new words contained in the neologism dictionary and the Google Play store descriptions.

4.4 Conversion Results

We demonstrate the validity of the resulting semantic vectors. For demonstration, we use the cosine similarity of the semantic vectors to measure the similarity between a pair of apps as Hoberg and Phillips (2016) did for comparing a pair of firms.

Table 2 and 3 present lists of apps sorted by the similarity of the app description to the target apps. The first row is the target app from which similarity is measured. The following rows contain apps whose similarity to the target app are ranked second to fifth highest, around the median, and the lowest. The numbers next to app names represent the cosine similarities to the target app.

Table 2: The similarity of the top application apps evaluated by NWJC2Vec

Order	Name of App	Similarity	Category
1	LINE: Free Calls & Messages	1.000	Communication
2	KakaoTalk: Free Calls & Text	0.991	Communication
3	Free live coverage and free phone calls an...	0.985	Social
4	MixChannel	0.985	Social
5	Facebook	0.983	Social
264	[Moba 7] Pachislot Monster Hunter late thu...	0.956	Casino
265	1010! Block Puzzle Game	0.955	Puzzle
266	Valkyrie connect	0.955	Role Playing
267	Dragon Quest VIII: cursed the sky and the ...	0.955	Role Playing
268	Power Battery - Battery Life Saver & Healt...	0.955	Tools
527	CR Eva X	0.874	Casino
528	Tower of Hero	0.866	Role Playing
529	I'm Juggler EX	0.825	Casino
530	nicoid (smiling video player)	0.807	Video Players & Editors
531	LIMBO	0.739	Adventure

Note: The first row is the target app from which similarity is measured. The following rows contain apps whose similarity to the target app are ranked second to fifth highest, around the median, and the lowest. The numbers next to app names represent the cosine similarities to the target app.

In Table 2, we select the most downloaded application app in our observations, *LINE: Free Calls & Messages*. The most similar apps to LINE are *KakaoTalk: Free Calls & Text* and *Free live coverage and free phone calls*. Both apps provide texts, images, voice, and video exchange services, as does LINE. *MixChannel*, a video sharing app, and *Facebook*, a social networking app, follow. Although they are listed in different product categories in Google Play, the cosine similarity of the semantic vectors successfully detected similar apps. The most different apps include Game apps, *Power Battery*, a battery-saving and cleaning app, and *nicoid*, a video player app for an anonymous video sharing service.

In Table 3, we included the most popular game app offered by the LINE's same developer, a puzzle game featuring Disney characters. All of the most similar apps are provided by the developer of LINE. *LINE: Bubble 2*, *LINE Pokopang*, and *LINE Puzzle TanTan* are all puzzle games. *Puyo!! Quest* is a puzzle game offered by a different developer. All of these five apps are puzzle games of a similar type. The median similarity apps and the most different apps fall into a variety of categories, such as News & Magazines, Tools, and Games. Apps belonging to the Games category do not contain a puzzle game of a similar type. Thus, the semantic vector successfully detect similar and different apps by nature. We studied a variety of other apps and obtained intuitive outputs.

To further validate the semantic vectors, we calculated the average cosine similarity within and between product categories. Table 4 provides the average similarity between a pair of entire apps, apps in different categories, and apps in the same category. The pair of same category apps are

Table 3: The similarity of the top game apps evaluated by NWJC2Vec

Order	Name of App	Similarity	Category
1	LINE: Disney Tsumutsumu	1.000	Puzzle
2	LINE Bubble 2	0.965	Puzzle
3	LINE Pokopang	0.964	Puzzle
4	LINE Puzzle TanTan	0.964	Puzzle
5	Puyo !! Quest - a large chain with a simpl...	0.962	Puzzle
264	Girl channel - Women's News and Girl Talk	0.928	News & Magazines
265	SimCity BuildIt	0.928	Simulation
266	My girlfriend is not something affair	0.928	Adventure
267	Geki J Pachi 2027	0.927	Casino
268	TRILL (tolyl) - Women's hair, fashion, coo...	0.927	News & Magazines
527	Free QR Scanner: Bar Code Scanner & QR Cod...	0.828	Tools
528	ChMate	0.817	Social
529	I'm Juggler EX	0.796	Casino
530	nicoid (smiling video player)	0.754	Video Players & Editors
531	LIMBO	0.721	Adventure

Note: The first row is the target app from which similarity is measured. The following rows contain apps whose similarity to the target app are ranked second to fifth highest, around the median, and the lowest. The numbers next to app names represent the cosine similarities to the target app.

statistically significantly more similar than the pair of different category apps (one-tailed t test statistics is 140.265). In addition, the minimum and median similarities are also smaller within a category than between categories. Thus, the semantic vector successfully captures similarities and differences embodied in the Google Play product categories.

4.5 Dimension Reduction

The resulting semantic vector has 300 dimensions. In the estimation, we consider low-dimensional and high-dimensional specifications. In the low-dimensional specification, we only include five

Table 4: Summary statistics of cosine similarity between different category apps

	Category	N	Mean	SD	Median	Min	Max
1	Overall	4613203	0.916	0.049	0.927	0.233	1
2	Pair of Different Category Apps	4387876	0.915	0.048	0.926	0.233	1
3	Pair of Same Category Apps	225327	0.930	0.051	0.943	0.337	1

Note: For the computations, we randomly draw 10% of the 30,357 apps and calculate the cosine similarity among them.

Table 5: Ratio of explained variance

	PC1	PC2	PC3	PC4	PC5
Individual	0.181	0.109	0.056	0.049	0.040
Cumulative	0.181	0.289	0.345	0.393	0.433

Note: Five dimensional principal components (PC) explain the variance of the product characteristics.

principal components of the semantic vector. In the high-dimensional specification, we include all dimensions of the semantic vector, and pick up relevant dimensions using a rigorous lasso (Belloni and Chernozhukov, 2013). Table 5 provides the explained variance by each the principal components. Five dimensional principal components explain 43.4% of the variance of the semantic vectors.

5 Model

In this section, we present a model of consumer's choice for mobile apps and app developer's pricing and non-pricing competition. The term *market* in this section means the sets of all apps at a time, and differs from a *relevant market* that is constructed for making antitrust policy decisions. In this section, we suppress the index of a market.

5.1 Setting

Population and covariates Consider a market with a set of apps $\mathcal{J} \coloneqq \{1, \dots, J\}$ provided by a group of app developers $\mathcal{D} \coloneqq \{1, \dots, D\}$. For each app, the developer can set the download price $F_j \in \mathbb{R}_+$ and in-app advertising intensity $a_j \in \mathbb{R}_+$. The market has a unit mass of consumers in the market. Each consumer has a unit download demand and decides on the app to download, how much to use the app, $q_j \in \mathbb{R}_+$, and how much to spend on the in-app purchases when using the app, $e_j \in \mathbb{R}_+$. Let w be the opportunity cost of a unit time for a consumer, that is, the wage.

When analyzing mobile apps, distinguishing utilities from an app's foreground and background processes of an app is important because the former requires consumers to spend their time, whereas the latter does not. For example, playing a game usually requires consumers to open and manually control the app. In contrast, an anti-virus software runs in the background and consumers only have to spend some time setting up the app after downloading it. In the following, we refer to the utility from a foreground process as *usage-related utility* and the utility from a background process as *download-related utility*, and model them separately. A usage-related utility should be a function of usage time whereas the download-related utility should be independent of usage time.

Let $X_{uj} \in \mathbb{R}^{K_u}$ and $X_{dj} \in \mathbb{R}^{K_d}$ be the observed characteristics of the app that affect a consumer's usage- and download-related utilities of a consumer. Let $\xi_{uj} \in \mathbb{R}$ and $\xi_{dj} \in \mathbb{R}$ be the characteristics of the app that affect a consumer's usage- and download-related utilities of a consumer but that are not observed to an econometrician. We assume that ξ_{uj}, ξ_{dj}, X_{uj}, and X_{dj} are mutually independent and $\mathbb{E}\{\xi_{uj}\} = \mathbb{E}\{\xi_{dj}\} = 0$.

Consumer preference The indirect utility from downloading and using app j for consumer i, u_{ij}, consists of usage-related and download-related components as follows:

$$u_{ij} \coloneqq S_j + \beta_{di}' X_{dj} - \alpha_y F_j + \xi_{dj} + \varepsilon_{ij} \tag{1}$$

where

$$S_j \coloneqq \max_{q_j, e_j} \{v_j(q_j, e_j, a_j, w, X_{uj}, \xi_{uj})\} \tag{2}$$

is the benefit from the optimal usage choice. The benefit from usage is assumed to have the following functional form:

$$v_j(q_j, e_j, a_j, X_{uj}, \xi_{uj}) \coloneqq [\beta_u' X_{uj} - \alpha_{aj} a_j - \alpha_y w + g_j(e_j, q_j) + \xi_{uj}] q_j - \psi_j(q_j) - \alpha_y e_j, \tag{3}$$

where ε_{ij} is an idiosyncratic taste shock distributed according to an i.i.d. type-I extreme-value distribution. We allow for β_{di} to have random coefficients as:

$$\beta_{di} \coloneqq \beta_d + \Sigma \nu_i, \tag{4}$$

with a K_d-dimensional random variable ν_i each of whose elements is drawn from an i.i.d. standard normal distribution. $\beta_u \in \mathbb{R}^{K_u}$ and $\beta_{di} \in \mathbb{R}^{K_d}$ represent the consumer's tastes for the characteristics, $\alpha_y \in \mathbb{R}_+$ is the utility from money, and α_{aj} is the disutility from being revealed to a unit advertisement in app j. We allow α_{aj} to vary across apps depending on observable characteristics and specify the form of α_{aj} as follows:

$$\alpha_{aj} = \alpha_{a1} \frac{\exp(\alpha_{a2}' X_{\alpha_a j})}{1 + \exp(\alpha_{a2}' X_{\alpha_a j})}, \tag{5}$$

where $X_{\alpha_a j} \in \mathbb{R}^{K_{\alpha_a}}$ is observed characteristics of the app that affect the disutilitiy of advertisements, and $\alpha_a \coloneqq (\alpha_{a1}, \alpha_{a2})$ is the parameter that determines the value of α_{aj}. We do not allow for the other parameters to have consumer-specific random coefficients because of a computational issue that we explain in detail in the relevant section. We expect that the inclusion of random coefficients in β_{di} should already allow for a flexible substitution pattern across apps.

Additional functional-form assumptions To obtain an analytical solution and facilitate computations while maintaining flexibility, we specify the functional forms of g_j and ψ as follows:

$$g_j(e_j, q_j) \coloneqq \sqrt{\xi_{ej} e_j / q_j}, \tag{6}$$

$$\psi_j(q_j) \coloneqq \frac{\eta_j}{2} q_j^2, \tag{7}$$

where $\xi_{ej} \in \mathbb{R}_+$ represents the characteristics of the app that affect the usage utility of a consumer through in-app purchases but is not observed to an econometrician. $\eta_j \in \mathbb{R}_+$ is the degree of satiation from usage, which is specified as follows:

$$\eta_j = \eta_1 \frac{\exp(\eta_2' X_{\eta j})}{1 + \exp(\eta_2' X_{\eta j})} + 0.05, \tag{8}$$

where $X_{\eta j} \in \mathbb{R}^{K_\eta}$ is observed characteristics of the app that affect the degree of satiation, and $\eta \coloneqq (\eta_1, \eta_2)$ is the parameter that determines the value of η_j. Because the model becomes numerically unstable as η_j approaches 0, we put a lower bound on it by adding 0.05. The estimate shows that this lower-bound is not binding.

As a result, the benefit from the optimal usage choice takes the following form:

$$S_j = \max_{q_j, e_j} \left\{ \left(\beta_u' X_{uj} - \alpha_{aj} a_j - \alpha_y w + \sqrt{\frac{\xi_{ej} e_j}{q_j}} + \xi_{uj} \right) q_j - \frac{\eta_j}{2} q_j^2 - \alpha_y e_j \right\}, \tag{9}$$

and the indirect utility takes the following form:

$$\begin{aligned} u_{ij} &= S_j - \alpha_y F_j + \beta_{di}' X_{dj} + \xi_{dj} + \varepsilon_{ij} \\ &\coloneqq \delta_j + \nu_i' \Sigma X_{dj} + \varepsilon_{ij}, \end{aligned} \tag{10}$$

where δ_j is the common mean indirect utility of consumers from app j.

5.2 Consumer's Problem

Usage and in-app purchase decisions Next, we solve the consumer problem. Let $X_j = (X_{uj}', X_{dj}')'$, $\xi_j = (\xi_{uj}, \xi_{dj}, \xi_{ej})$, $\theta_i = (\alpha_{aj}, \beta_u', \alpha_y, \eta_j, \beta_{di}')'$, and $\theta = (\alpha_{aj}, \beta_u', \alpha_y, \eta_j, \beta_d', \mathrm{vec}(\Sigma)')'$. By solving the first-order condition for in-app purchase e_j, we obtain the following relationship between in-app purchase and usage:

$$e_j = \frac{1}{4\alpha_y^2} \xi_{ej} q_j. \tag{11}$$

By solving the first-order condition for usage q_j and inserting equation (11), we obtain:

$$q_j = \tilde{q}_j(a_j, w, X_j, \xi_j; \theta) \coloneqq \max \left\{ \frac{1}{\eta_j} \left(\beta_u' X_{uj} - \alpha_{aj} a_j - \alpha_y w + \frac{\xi_{ej}}{2\alpha_y} + \xi_{uj} \right), 0 \right\}. \tag{12}$$

By inserting this back into equation (11), we obtain:

$$e_j = \tilde{e}_j(a_j, w, X_j, \xi_j; \theta) \coloneqq \frac{\xi_{ej} \tilde{q}_j(a_j, w, X_j, \xi_j; \theta)}{4\alpha_y^2}. \tag{13}$$

By substituting equation (12) into equation (9), we obtain the following usage surplus function:

$$S_j = S(a_j, w, X_j, \xi_j; \theta) \coloneqq \frac{\eta_j}{2} \tilde{q}_j^2(a_j, w, X_j, \xi_j; \theta) - \alpha_y \tilde{e}_j(a_j, w, X_j, \xi_j; \theta), \tag{14}$$

which leads to the mean indirect utility δ_j of consumers from app j:

$$\begin{aligned}\delta_j =&\delta_j(a_j, F_j, w, X_j, \xi_j; \theta)\\ \coloneqq&S(a_j, w, X_j, \xi_j; \theta) + \beta_d' X_{dj} - \alpha_y F_j + \xi_{dj}.\end{aligned} \tag{15}$$

Download decision Next, we derive the probability that a consumer downloads an app. Let $a = (a_j)_{j\in\mathcal{J}}$, $F = (F_j)_{j\in\mathcal{J}}$, $X = (X_j)_{j\in\mathcal{J}}$, w, and $\xi = (\xi_j)_{j\in\mathcal{J}}$. Under the assumption that ε_{ij} follows an i.i.d. type-I extreme-value distribution, the probability that a consumer downloads app j is:

$$s_j = \tilde{s}_j(a, F, w, X, \xi; \theta) \coloneqq \int_{\mathbb{R}^{K_d}} \frac{\exp\left[\delta_j(a_j, F_j, w, X_j, \xi_j; \theta) + \nu_i' \Sigma X_{dj}\right]}{1 + \sum_k \exp\left[\delta_k(a_k, F_k, w, X_k, \xi_k; \theta) + \nu_i' \Sigma X_{dk}\right]} dG_{\nu_i}(\nu_i). \tag{16}$$

5.3 Developer's Problem

Developer's profit Now consider an app developer's decisions related to its apps' download prices and advertising intensity. Let $\mathcal{J}_d \subset \mathcal{J}$ be the set of apps that developer d sells. A developer's profit is the sum of profits from each app, as follows:

$$\Pi_d(a, F, X, w, \xi; \theta) \coloneqq \sum_{j\in\mathcal{J}_d} \pi_j(a, F, X, w, \xi; \theta), \tag{17}$$

and the profit from each app consists of the revenues from downloads, in-app purchases, and advertisements:

$$\begin{aligned}&\pi_j(a, F, X, w, \xi; \theta)\\ \coloneqq&s_j(a, F, X, w, \xi; \theta)\left\{(1-\rho)[F_j + e_j(a, F, X, w, \xi; \theta)] + q_j(a, F, X, w, \xi; \theta)(a_j r - \lambda_j)\right\},\end{aligned} \tag{18}$$

where r is the advertising revenue per unit of advertisements shown to the consumer, ρ is the royalty rate that app developers pay to the app platform (i.e., Apple App Store and Google Play Store) for each download of the apps and the in-app purchases, and λ_j is other marginal cost. λ_j represents a constant marginal cost. We allow λ_j to vary according to the observable characteristics, and it is specified as follows:

$$\lambda_j = \lambda_1 \frac{\exp\left(\lambda_2' X_{\lambda j}\right)}{1 + \exp\left(\lambda_2' X_{\lambda j}\right)}, \tag{19}$$

where $X_{\lambda j} \in \mathbb{R}^{K_\lambda}$ represents the observed characteristics of the app that affect the marginal cost, and $\lambda \coloneqq (\lambda_1, \lambda_2)$ is the parameter that determines the value of λ_j.

Key identification assumption We imposed the following key identification assumption for our model: *no direct marginal cost of showing an advertisement on their app exists other than the loss from a decrease in demand attributable to the inconvenience caused to consumers by the advertisement.* We note that serving more consumers and greater usage incurs a marginal cost. Revenue is lost from decreasing consumer demand attributable to an increase in advertising intensity. What is assumed to be zero here is the *direct* marginal cost regarding increasing advertising intensity a_j.

In a standard merger analysis, we estimate the marginal cost from a firm's pricing decisions. However, in this paper, we estimate advertising intensity, the effective price, from the optimality condition assuming that no marginal cost exists that is specific to the decision. This assumption seems to be valid because connecting to ad networks and showing advertisement distributed through networks is almost automatic. The existing literature of ad-sponsored media focused on the media such as newspapers and cable TV. In their models, the costs of printing and producing advertisements, and acquiring new sponsors are included as direct marginal cost parameters of advertisements. They are not relevant in the context of mobile app advertisements.

Han et al. (2016) used a dummy for showing advertisements as one of the product characteristics of an mobile app. We use the same information, but in a different way. We use the advertising dummy as a partial observation of advertising intensity and match the dummy with elicited advertising intensity, as discussed in further detail in the estimation section. Remark that the identification comes from the assumption of no direct marginal cost of advertising and the advertisement dummy is used only to further discipline the estimates.

Download price and advertising intensity decisions The decision problem for app developer d is written as:

$$\max_{\{(a_j,F_j)\}_{j\in\mathcal{J}_d}} \Pi_d(a,F,X,\xi;\theta) \tag{20}$$

$$\text{s. t.} \quad a_j \geq 0, \quad j \in \mathcal{J}_d, \tag{21}$$

$$F_j \geq 0, \quad j \in \mathcal{J}_d. \tag{22}$$

The first-order conditions for this problem are:

$$\frac{\partial \Pi_d}{\partial F_j} := (1-\rho)s_j + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial F_j}[(1-\rho)(F_k+e_k) + q_k(a_k r - \lambda_k)] \leq 0, \tag{23}$$

with equality if $F_j > 0$ for each $j \in \mathcal{J}_d$, and:

$$\begin{aligned} \frac{\partial \Pi_d}{\partial a_j} := {} & s_j q_j r + s_j \frac{\partial q_j}{\partial a_j}(a_j r - \lambda_j) + s_j(1-\rho)\frac{\partial e_j}{\partial a_j} \\ & + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial a_j}\{(1-\rho)(F_k+e_k) + q_k(a_k r - \lambda_k)\} \\ & \leq 0, \end{aligned} \tag{24}$$

with equality if $a_j > 0$ for each $j \in \mathcal{J}_d$, where the component derivatives are:

$$\frac{\partial s_k}{\partial y} := \frac{\partial \tilde{s}_k(a, F, w, X, \xi; \theta)}{\partial y} \text{ for } y \in \{F_j, a_j\},$$
$$\frac{\partial q_k}{\partial a_j} := \frac{\partial \tilde{q}(a_j, w, X_j, \xi_j; \theta)}{\partial a_j},$$
$$\frac{\partial e_k}{\partial a_j} := \frac{\partial \tilde{e}(a_j, w, X_j, \xi_j; \theta)}{\partial a_j}.$$

A Bertrand-Nash equilibrium of the pricing game is a profile of pairs of advertising intensity and download prices $(a_j, F_j)_{j \in \mathcal{J}}$ that satisfies the system of equations (23) and (24).

5.4 Computing Equilibrium

Algorithm We propose an algorithm to numerically compute an equilibrium of the model. We show that this algorithm converges to the least equilibrium. Specifically, let $\Psi_d(a, F)$ be the best-response function of app developer d and define the best-response mapping Ψ by

$$\Psi(a, F) := \{\Psi_1(a, F), \dots, \Psi_D(a, F)\}.$$

Then, we compute an equilibrium using the following iteration algorithm.

1. Step 0: Set the initial value by (a^0, F^0) by $(a_j^0, F_j^0) := (0, 0)$ and update $(a^1, F^1) = \Psi(0, 0)$.
2. Step k: Given (a^k, F^k), update $(a^{k+1}, F^{k+1}) = \Psi(a^k, F^k)$.

In super-modular games, an iteration algorithm that starts from the least element of a strategy space converges to the least equilibrium. In our model, although the game is not super-modular, it turns out that if the heterogeneity in the coefficient on download-related product characteristics β_{dj} is not too large, the iteration algorithm converges to the least equilibrium because the best-response mapping is monotone increasing in our model. Together with the continuity of the best-response mapping, this monotonicity guarantees that the algorithm converges to the least equilibrium in the model. See Appendix A for the proof. In the implementation of the iteration algorithm, to improve the numerical performance, we impose the constraint that $(a^k, F^k) \geq (a^{k-1}, F^{k-1})$ to during the computation of app-developers' best-response in each step k to improve the numerical performance. In theory, this constraint does not affect the iteration procedure as long as the best-response functions are increasing. In practice, this constraint significantly improves the stability of the iteration algorithm.

5.5 Monte Carlo simulations

Figure 5 shows the convergence property of the previously introduced iteration algorithm. For the simulation, we consider the environment with 50 app developers, and each developer provides two apps. To observe the impact of consumer heterogeneity on the convergence property of the

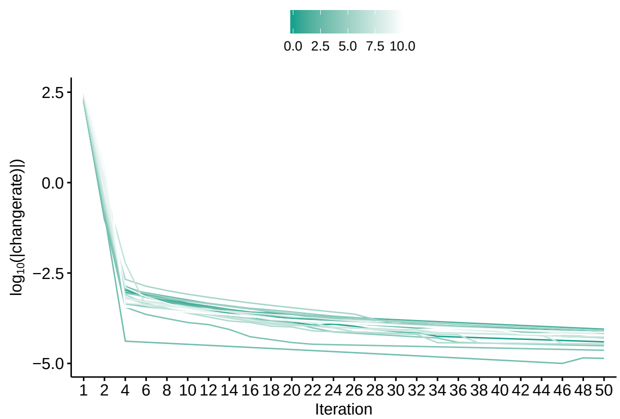


Figure 5: Convergence of iteration algorithm

iteration algorithm, we set $K_d = 6$, draw non-constant elements of X_{dj} from uniform distribution on $[0,1]^5$, and draw σ from uniform distribution on $[0,1]^6$. In Figure 5, we vary σ by multiplying constants ranging from 0 to 10. The horizontal axis represents the steps of iterations k, and the vertical axis represents the common logarithm of the distance between (a^k, F^k) and (a^{k-1}, F^{k-1}) which is defined by

$$\max\left\{\max_j\left\{\frac{|a_j^k - a_j^{k-1}|}{1+|a_j^{k-1}|}\right\}, \max_j\left\{\frac{|F_j^k - F_j^{k-1}|}{1+|F_j^{k-1}|}\right\}\right\}.$$

For each scale, the trajectory of the distances starts from a large value, soon approaches $10^{-2.5}$, and then further diminishes. This pattern shows that our iteration algorithm performs well in terms of the convergence.

Tables 13-15 of Appendix C illustrate how the equilibrium prices and advertising intensities depend on the underlying model parameters.

6 Estimation

6.1 Moment Conditions for Consumer Choice with Advertising Elicitation

We fix the parameters and data and first solve the equilibrium conditions for unobserved fixed effects ξ_{ej}, ξ_{uj}, and ξ_{dj}. To solve for ξ_{uj}, we elicit the advertising intensity a_j that is implied from the parameters and the data. Then, we define a generalized method-of-moments estimator that exploits the moments regarding these unobserved fixed effects. Let $\theta := (\theta_d', \theta_u', lambda')'$, where $\theta_d := (\alpha_y, \eta, \beta_d', \mathrm{vec}(\Sigma)')'$ is a set of parameters related to the download-related moment condition and $\theta_u := (\alpha_a', \beta_u')$ is the set of parameters. Let $\theta_0 := (\theta_{d0}', \theta_{u0}', \lambda_0')'$ denote true parameters.

Solving for ξ_e By arranging the first-order condition for a consumer regarding her in-app purchase decision (11), we can have:

$$\xi_{ej} = 4\alpha_y^2 \frac{e_j}{q_j}. \tag{25}$$

Let $\xi_{ej}(\theta_d)$ be the implied value of ξ_{ej} for $j \in \mathcal{J}$.

Solving for ξ_d We solve for the value of ξ_d from the following equation:

$$s_j = \int_{\mathbb{R}^{K_d}} \frac{\exp\left[\delta_j(a_j, F_j, w, X_j, \xi_j; \theta) + \nu_i' \Sigma X_{dj}\right]}{1 + \sum_k \exp\left[\delta_k(a_k, F_k, w, X_k, \xi_k; \theta) + \nu_i' \Sigma X_{dk}\right]} dG_{\nu_i}(\nu_i). \tag{26}$$

Although this equation involves ξ_{ej} and ξ_{uj} in general, we can solve for the values of ξ_{dj} as a function of observable variables and parameters. We do so by using the following equation:

$$\begin{aligned} \delta_j(a_j, F_j, w, X_j, \xi_j; \theta) &= S(a_j, w, X_j, \xi_j; \theta) + \beta_d' X_{dj} - \alpha_y F_j + \xi_{dj} \\ &= \frac{\eta_j}{2} q_j^2 - \alpha_y e_j + \beta_d' X_{dj} - \alpha_y F_j + \xi_{dj}. \end{aligned} \tag{27}$$

By inserting equation (27) into equation (26), we can express the share equation in terms of the parameters, observables, and values of ξ_{dj}. Then, we compute the implied value of ξ_d through a BLP-type inversion (Berry et al., 1995). Let $\xi_d(\theta_d)$ denote the implied values because the equation only depends on θ_d, given that the dependence of S on a_j, w, X_{uj}, ξ_{uj} works only through q_j and e_j in equation (27). This dependence results from the functional-form assumption in (12), a trick that allows us to separate the elicitation of ξ_d and ξ_u and substantially facilitates computation.

Additionally, note that this argument works because we did not allow random-coefficients for usage-related indirect utility. If the coefficients of X_{uj} were stochastic across consumers, then q_j in equation (27) would have been stochastic across consumers and indexed as q_{ij}. If we had consumer-level usage data, we could estimate a distribution of q_{ij} and integrate q_{ij} out from equation (26) under the condition that the conditional distribution of q_{ij} on that app j is chosen is the same as its unconditional distribution. The latter condition holds if the random coefficients on X_{dj} and X_{uj} are independent and the random coefficients on X_{uj} are realized after the consumer actually downloads the app.

Because we do not have consumer-level data, we cannot follow this approach. Then, allowing for random coefficients on X_{uj} requires us to solve the distribution of q_j under candidate parameters to evaluate an objective function of an estimator. This requirement significantly complicates the computational task. We stress that these restrictions on unobserved heterogeneity and functional form are utilized primarily to facilitate computation but not for identification.

Solving for a Next, we elicit advertising intensity $\{a_j\}_{j\in\mathcal{J}}$. To elicit the advertising intensity $\{a_j\}_{j\in\mathcal{J}}$, we utilize the first-order conditions for advertising intensity:

$$s_j q_j r + s_j \frac{\partial q_j}{\partial a_j}(a_j r - \lambda_j) + s_j(1-\rho)\frac{\partial e_j}{\partial a_j} + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial a_j}\{(1-\rho)(F_k + e_k) - \lambda_k + q_k a_k r\} \leq 0, \tag{28}$$

where

$$\frac{\partial q_j}{\partial a_j} = -\frac{\alpha_{aj}}{\eta_j}, \frac{\partial e_j}{\partial a_j} = -\frac{\alpha_{aj} q_j \xi_{ej}}{4\alpha_y^2 \eta_j}, \tag{29}$$

and

$$\frac{\partial s_k}{\partial a_j} = \begin{cases} -\alpha_{aj} q_j \int_{\mathbb{R}^{K_d}} \frac{\exp(\delta_k + \nu_i' \Sigma X_{dk'})}{\left[1+\sum_{k'} \exp(\delta_k + \nu_i' \Sigma X_{dk'})\right]} \left[1 - \frac{\exp(\delta_k + \nu_i' \Sigma X_{dk'})}{\left[1+\sum_{k'} \exp(\delta_k + \nu_i' \Sigma X_{dk'})\right]}\right] dG_{\nu_i}(\nu_i) & \text{for } k = j \\ -\alpha_{aj} q_j \int_{\mathbb{R}^{K_d}} \frac{\exp(\delta_j + \nu_i' \Sigma X_{dj}) \exp(\delta_k + \nu_i' \Sigma X_{dk'})}{\left[1+\sum_{k'} \exp(\delta_k + \nu_i' \Sigma X_{dk'})\right]^2} dG_{\nu_i}(\nu_i) & \text{for } k \neq j \end{cases} \tag{30}$$

for all $j \in \mathcal{J}$. Given the data $(s_j, q_j, p_j)_{j\in\mathcal{J}}$ and computed values of $(\xi_{ej}(\theta), \xi_{dj}(\theta))_{j\in\mathcal{J}}$, we can compute the simulated value of $\partial s_k / \partial a_j$. Solving this system of equations and inequalities is complicated in general. However, with our model setting, we can compute the values of advertising intensity $(a_j)_{j\in\mathcal{J}}$ that satisfy the system of first-order conditions (28) by solving quadratic programming. The details of the procedure and the proof are in the Appendix B. Let $a(\theta)$ denote the elicited advertising intensity.

Solving for ξ_u By plugging $\xi_e(\theta_d)$ and $a(\theta)$ into the first-order condition for usage (12), we obtained the implied value of ξ_{uj}:

$$\xi_{uj} = \eta_j q_j + \alpha_{aj} a_j(\theta) + \alpha_y w - \beta_u' X_{uj} - \frac{\xi_{ej}(\theta_d)}{2\alpha_y}. \tag{31}$$

Let $\xi_{uj}(\theta)$ be the implied value of ξ_{uj} for each $j \in \mathcal{J}$.

Moment conditions Let $Z_{uj} \in \mathbb{R}^{L_u}$ and $Z_{dj} \in \mathbb{R}^{L_d}$ are sets of instrumental variables for app j that satisfy:

$$\mathbb{E}[\xi_{uj}(\theta_0)|Z_{uj}] = \mathbb{E}[\xi_{dj}(\theta_{d0})|Z_{dj}] = 0, \tag{32}$$

which implies:

$$\mathbb{E}[\xi_{uj}(\theta_0) Z_{uj}] = \mathbb{E}[\xi_{dj}(\theta_{d0}) Z_{dj}] = 0, \tag{33}$$

for any $j \in \mathcal{J}$.

Objective Function Let $t \in \mathcal{T} := \{1, \ldots, T\}$ be the set of indices of markets, $\mathcal{J}_t$ the set of apps in market t, and $(X_{ujt}, X_{djt}, Z_{ujt}, Z_{djt}, e_{jt}, F_{jt}, q_{jt}, s_{tj})$ the list of variables regarding app j in market t. Let $N \coloneqq \sum_{t=1}^{T} J_t$. Let:

$$\xi_{ut}(\theta) \coloneqq [\xi_{u1t}(\theta), \cdots, \xi_{uJ_tt}(\theta)]', \xi_{dt}(\theta_d) \coloneqq [\xi_{d1t}(\theta_d), \cdots, \xi_{dJ_tt}]', \tag{34}$$

be the J_t-dimensional vector of product-specific unobserved heterogeneity in market t and:

$$\xi_u(\theta) \coloneqq [\xi_{u1}(\theta)', \cdots, \xi_{uT}(\theta)']', \xi_d(\theta_d) \coloneqq [\xi_{d1}(\theta_d)', \cdots, \xi_{dT}(\theta_d)']', \tag{35}$$

be the $\sum_{t=1}^{T} J_t$-dimensional vector of product-market-specific unobserved heterogeneity.

Similarly, for instrumental variables and product characteristics, let:

$$Z_{\iota t} \coloneqq \begin{pmatrix} Z'_{\iota 1t} \\ \vdots \\ Z'_{\iota J_t t} \end{pmatrix}, X_{\iota t} \coloneqq \begin{pmatrix} X'_{\iota 1t} \\ \vdots \\ X'_{\iota J_t t} \end{pmatrix}, \iota \in \{u, d\}, \tag{36}$$

be the $J_t \times L_\iota$-dimensional matrix of instrumental variables and the $J_t \times K_\iota$-dimensional matrix of instrumental variables in market t and:

$$Z_\iota \coloneqq \begin{pmatrix} Z_{\iota 1} \\ \vdots \\ Z_{\iota T} \end{pmatrix}, X_\iota \coloneqq \begin{pmatrix} X_{\iota 1} \\ \vdots \\ X_{\iota T} \end{pmatrix}, \iota \in \{u, d\}, \tag{37}$$

be $\sum_{t=1}^{T} J_t \times L_\iota$-dimensional matrix of instrumental variables and $\sum_{t=1}^{T} J_t \times K_\iota$-dimensional matrix of product characteristics.

Finally, let:

$$g^D(\theta) \coloneqq \frac{1}{\sum_{t \in \mathcal{T}} J_t} \begin{pmatrix} Z'_u \xi_u(\theta) \\ Z'_d \xi_d(\theta_d) \end{pmatrix} \tag{38}$$

be the $L_u + L_d$-dimensional moments related to the demand with elicited advertising.

6.2 Moment Conditions for App Developer's Choice

Optimality conditions for download price For each $j \in \mathcal{J}_t$ in each $t \in \mathcal{T}$, the following equality holds:

$$\epsilon^P_{jt}(\theta) \coloneqq \frac{\partial \Pi_{jt}(\theta)}{\partial F_{jt}} 1\{F_{jt} > 0\} + \max\left\{\frac{\partial \Pi_{jt}(\theta)}{\partial F_{jt}}, 0\right\} 1\{F_{jt} = 0\} = 0. \tag{39}$$

We construct a corresponding moment such as:

$$g^P(\theta) \coloneqq \frac{1}{\sum_{t \in \mathcal{T}} J_t} Z'_u \epsilon^P(\theta), \tag{40}$$

where $\epsilon^P(\theta) \coloneqq [\epsilon^P_{11}(\theta), \cdots, \epsilon^P_{J_T T}(\theta)]'$.

Advertising matching Although advertising intensity is not observed, we observe whether or not an app shows advertisements or not. Given the true parameter, we expect that approximately the following equation holds:

$$\epsilon_{jt}^{A}(\theta) := [A_{jt} - 1\{a_{jt}(\theta) > 0\}] = 0, \tag{41}$$

where A_{jt} takes the value of 1 if app j shows advertisements in market t and takes the value of 0 otherwise. We construct a corresponding moment such as:

$$g^{A}(\theta) := \frac{1}{\sum_{t \in \mathcal{T}} J_t} Z_u' \epsilon^{A}(\theta), \tag{42}$$

where $\epsilon^{A}(\theta) := [\epsilon_{11}^{A}(\theta), \cdots, \epsilon_{J_T T}^{A}(\theta)]'$.

6.3 Generalized Method-of-Moments Estimator

Definition We define a generalized method-of moments (GMM) estimator $\hat{\theta}$ by:

$$\hat{\theta} \in \operatorname{argmin}_{\theta \in \Theta} g(\theta)' \Phi^{-1} g(\theta), \tag{43}$$

where $g(\theta) := [g^{D\prime}(\theta), g^{P\prime}(\theta), g^{A\prime}(\theta)]'$ and Φ is a positive-definite weighting matrix.

We start with an initial weighting matrix blkdiag$[Z_u' Z_u, Z_d' Z_d, Z_u' Z_u, Z_u' Z_u]$. Then, in the second step, we use the sample covariance of the moments evaluated at the initial estimates. We first estimate a model without random coefficients nor heterogeneity in $\alpha_{aj}, \eta_j, \lambda_j$. Then, we estimate the model by first adding the random coefficients and, second, the heterogeneity, using the previous estimates as the initial values. Because of the computational burden, we estimate the parameters by randomly sub-sampling 20 markets (weeks) from the entire data. This sub-sampling provides approximately 10,000 observations at the app-market level. We obtain the confidence intervals by repeatedly estimating the parameters using a randomly selected list of sub-samples. We minimize the objective function by using an adaptive barrier algorithm implemented by the *constrOptim* function in R to impose non-negativity constraints on the parameters except for the parameters governing the heterogeneity of $\alpha_{aj}, \eta_j, \lambda_j$.

Choice of instrumental variables Z_{djt} includes 1, X_{djt}, X_{djt}^2 and differentiation instrumental variables (Gandhi and Houde, 2019). Specifically, for each app, for $\iota \in \{d, u\}$, compute the difference from the other apps in the product characteristics space:

$$d_{\iota jkt} := \sqrt{\sum_{l=1}^{K_\iota} (X_{\iota jlt} - X_{\iota klt})^2}, \tag{44}$$

and compute the average and variance of the differences within the same app developer and outside the app developer:

$$\frac{1}{J_{d(j)t}} \sum_{k \in \mathcal{J}_{d(j)t}} d_{\iota jkt}, \frac{1}{J_t - J_{d(j)t}} \sum_{k \in \mathcal{J}_t \setminus \mathcal{J}_{d(j)t}} d_{\iota jkt}, \frac{1}{J_t - 1} \sum_{k \in \mathcal{J}_t} \left(d_{\iota jkt} - \frac{1}{J_t} \sum_{k \in \mathcal{J}_t} d_{\iota jkt} \right)^2 . \tag{45}$$

Moreover, we include hourly wage and advertising price as market-level demand and cost shifters. Z_u includes the corresponding variables except for X_{ujt}^2, because there is no random-coefficient in the usage-related utility.

Linear and non-linear parameters We can further accelerate the computation by distinguishing between *linear* and *non-linear* parameters; linear parameters can be explicitly derived by minimizing the objective function in equations (42), given the rest of the parameters. Specifically, the linear parameters in θ_d and θ_u in our framework are $\theta_{d1} := \beta_d$ and $\beta_{u1} := \beta_u$, and the non-linear parameters in θ_d and θ_u in our framework are $\theta_{d2} := [\alpha_y, \eta_j, \mathrm{vec}(\Sigma)']'$ and $\theta_{u2} := \alpha_{aj}$. Given $\theta_2 := (\theta_{2d}', \theta_{2u}', \lambda_j)'$, the residuals in the demand-related moment condition are written as:

$$\begin{aligned} \xi_d(\theta) &= y_d(\theta) - X_d \beta_d, \\ \xi_u(\theta) &= y_u(\theta) - X_u \beta_u, \end{aligned} \tag{46}$$

with:

$$\begin{aligned} y_d(\theta) &:= \delta(\theta) - \frac{\eta_j}{2} q^2 + \alpha_y e + \alpha_y F, \\ y_u(\theta) &:= \eta_j q + \alpha_{aj} a(\theta) + \alpha_y w - \frac{\xi_e(\theta_d)}{2\alpha_y}, \end{aligned} \tag{47}$$

where $\delta(\theta), q, e, F, a(\theta), w$, and $\xi_e(\theta_d)$ are vectors in which corresponding elements are stacked first by apps and then by markets.

Both $y_d(\theta)$ and $y_u(\theta)$ depend on θ_1 through $\delta(\theta)$, $a(\theta)$, and $\xi_e(\theta_d)$. However, in our specification of the model, $\delta(\theta)$, $a(\theta)$, and $\xi_e(\theta_d)$ are independent of θ_1 conditional on the observables $(s, q, e, X_d) = \{(s_j, q_j, e_j, X_{dj})_{j \in \mathcal{J}_t}\}_{t=1,\ldots,T}$ and non-linear parameters θ_2 for the following reasons. First, equation (26) implies that $\delta(\theta)$ can be computed only using (s, X_d), denoted by $\hat{\delta}(\theta_2, s, X_d)$. Similarly, equation (25) implies that $\xi_e(\theta_d)$ can be computed only using q, e, and α_y, denoted by $\hat{\xi}_d(\theta_{2d}, q, e)$. Finally, equations (28), (29), and (30) jointly imply that $a(\theta)$ can be computed only using variables (s, q, e, X_d), $\xi_e(\theta_d)$, $\delta(\theta)$ and non-linear parameters θ_2, denoted by $\hat{a}(\theta_2, s, q, e, X_d)$. Therefore, $y_d(\theta)$ can be evaluated as $\hat{y}_d(\theta_2, s, q, e, F, X_d)$ and $y_u(\theta)$ can be evaluated as $\hat{y}_u(\theta_2, s, q, e, w, X_d)$. Similarly, $g^P(\theta)$ can be evaluated by (s, q, e, F, r), λ_j, and $\hat{a}(\theta_2, s, q, e, X_d)$, denoted by $\widehat{g}^P(\theta_2, s, q, e, F, r, X_d)$. Finally, $g^A(\theta)$ is evaluated only with $\hat{a}(\theta_2, s, q, e, X_d)$, denoted by $\widehat{g}^A(\theta_2, s, q, e, X_d)$.

As a result, given observables and fixed non-linear parameters, we can ignore the impact of linear parameters on $y_d(\theta)$ and $y_u(\theta)$, and $g^P(\theta)$ and $g^A(\theta)$, which enables us to explicitly derive the estimates of linear parameters conditional on non-linear parameters as follows:

$$\hat{\theta}_1(\theta_2) = \left(X'Z\Phi^{D-1}Z'X\right)^{-1} X'Z\Phi^{D-1}Z'\hat{y}(\theta_2, s, q, e, F, r, w, X_d), \tag{48}$$

where $\hat{y}(\theta_2, s, q, e, F, r, w, X_d) := [\hat{y}_d(\theta_2, s, q, e, F, X_d)', \hat{y}_u(\theta_2, s, q, e, w, X_d)']'$, $Z := \text{blkdiag}(Z_u, Z_d)$, and $X := \text{blkdiag}(X_u, X_d)$. Φ^D is a submatrix of Φ corresponding to demand-releated moments.

6.4 Incorporating Semantic Vectors of Product Description

Semantic vectors without random coefficients The product description of an app is represented by a semantic vector, which we denote by $W_j \in \mathbb{R}^P$. Product attributes encoded in W_j surely affect consumer demand; however, which of them will do so is not *a priori* clear. Therefore, we allow data to indicate the dimension of W_j that is particularly relevant. The interpretation of X_{dj}, X_{uj}, and W_j is that X_{dj} and X_{uj} are variables that certainly affect utility, and W_j represents variables with uncertain influence. First, we assume that no consumer-level heterogeneity exists regarding the coefficients for W_j. If this is the case, W_j should be part of the unobserved heterogeneity ξ_{dj} and ξ_{uj} in the previous model:

$$\begin{aligned} \xi_{dj} &= \gamma_d' W_j + \Delta\xi_{dj}, \\ \xi_{uj} &= \gamma_u' W_j + \Delta\xi_{uj}, \end{aligned} \tag{49}$$

where $\Delta\xi_{dj}$ and $\Delta\xi_{uj}$ represent residual unobserved heterogeneity that is not correlated with W_j. Using fitted values $\hat{\xi}_{dj}(\hat{\theta})$ and $\hat{\xi}_{uj}(\hat{\theta})$ based on the GMM estimator $\hat{\theta}$, we can estimate γ_d and γ_u by regressing the fitted values on W_j. Because the semantic space of the in-text product description is high-dimensional, the ordinary least square estimates may over-fit to the training data. Therefore, to improve generalization performance, we estimate using rigorous post-LASSO estimators, in which the penalty loading of each variable is calculated depending on the variables and allowing for heteroskedasticity (Belloni and Chernozhukov, 2013).

Semantic vectors with random coefficients Next, we consider a model in which consumers have heterogeneous tastes for the features represented by the semantic vector of a mobile app in an unobserved manner. Then, the model is no different from the previous model in which X_{dj} and X_{uj} are replaced with $(X_{dj}', W_j)'$ and $(X_{uj}', W_j')'$, respectively. However, estimating this model is practically not possible, because too many parameters need to be estimated. Therefore, we adopt a short cut and we only use dimensions of W_j that are found to be relevant in the rigorous post-LASSO estimator assuming that no random coefficients exist as for the semantic vectors. Although this is an approximation does not have a rigorous theoretical background, we can expect that the dimension of a semantic vector with negligible first-order effects have negligible second-order effects.

7 Estimation Result

Estimates of non-linear and linear parameters Table 6 provides a summary of estimation results of non-linear parameters. In the high-dimensional heterogeneous mixed-logit specification, the estimate of α_y is 0.0001. Thus, the standard deviation in the download preference shock is huge. The estimate of α_{a1} is 0.1059. Because the unit of a_{jt} is 1000 counts per hour, an increase of one advertising per hour on baseline apps (casual game) annoys consumers as much as JPY5.3 $= \frac{0.1059}{2\times 1000\times 0.0001}$. The estimate of the satiation scale η_1 is 0.1552, which implies that if the equilibrium usage time of an app is one hour longer than another app, the marginal utility of usage of the app is $0.13 = \frac{0.1552}{2} + 0.05$ more than another app. The estimate of the marginal cost scale λ_1 is 0.1836. The random-coefficients have relatively small variations compared with idiosyncratic preference shocks, suggesting that apps that apparently belong to different categories may be close substitutes from a consumer's viewpoint.

Table 7 summarizes the estimation results of the linear parameters. Regarding the principal components of semantic vectors and product class dummies, gaming apps on average have a lower utility of download and a substantially higher marginal utility of usage. In contrast, tools that enhance productivity have a higher utility of download, but are not used for a long time. Shopping apps and apps for acquiring information have similar patterns. These patterns are intuitive. The coefficients on the principal component appear relevant for usage and download utilities as much as do product class dummies.

Model fit Although the model is highly stylized, it well fits the download and usage data. Figure 6 contrasts the download and usage data with the models fitted values. To obtain a fitted value, we need to take expectation with respect to the underlying shocks ξ_{dj}, ξ_{uj}, and ϵ_{ij}. Taking expectation with respect to idiosyncratic shocks ϵ_{ij} is straightforward, because they are assumed to be an i.i.d. type-I extreme random variable. However, we need a numerical integration for ξ_{dj} and ξ_{uj}. Specifically, we first regress $\hat{\xi}_{dj}(\hat{\theta})$ and $\hat{\xi}_{uj}(\hat{\theta})$ on endogenous variables a_j, p_j, and q_j using a generalized additive model, because they can be the unobserved heterogeneity can be correlated with those variables, and obtain residuals. For each observation, we draw samples of ξ_{dj} and ξ_{uj} by re-sampling the residuals and adding the expected value from the generalized additive model. For each draw of ξ_{dj} and ξ_{uj}, we solve consumer usage and download decisions and take an average across the samples to obtain the fitted value of download share and usage time. Table 8 shows that the R-squared of regressing the in-sample data on fitted values is 0.93 for download and 0.89 for usage. The R-squared of regressing the out-of-sample data on fitted values is 0.93 for download and 0.91 for usage. Thus, consumer behavior is sufficiently accurately explained by our model, both in-sample and out-of-sample.

Relevance of semantic vectors Figure 7 shows the estimates of γ_d and γ_u in descending order with the size of the absolute value of the estimates. Among the 300 dimensions, approximately 40-60 dimensions are picked up by the rigorous Post-Lasso method, which underscores the relevance

Table 6: Estimation result

	Description	Parameter	Low-dimensional Homogeneous	Low-dimensional Heterogeneous	High-dimensional Heterogeneous
1	Utility per JPY	α_y	1e-04	0.0001	0.0001
2	Utility per advertising (scale)	α_{a1}	0.1034	0.0969	0.1059
3	Hardcoregame	α_{a21}		-0.0015	-0.0015
4	Tools	α_{a22}		-0.0019	-0.0019
5	Shopping	α_{a23}		-0.0065	-0.0065
6	Information	α_{a24}		-0.0028	-0.0028
7	Degree of usage satiation (scale)	η_1	0.155	0.1552	0.1552
8	Hardcoregame	η_{21}		-0.0025	-0.0025
9	Tools	η_{22}		-0.0005	-0.0005
10	Shopping	η_{23}		-0.0012	-0.0012
11	Information	η_{24}		-0.0008	-0.0008
12	Marginal cost (scale)	λ_1	0.176	0.1836	0.1836
13	Hardcoregame	λ_{21}		0.0056	0.0056
14	Tools	λ_{22}		0.0250	0.0342
15	Shopping	λ_{23}		0.0167	0.0195
16	Information	λ_{24}		0.0056	0.0200
17	SD at intercept	σ_1	0.0108	0.0141	0.0141
18	SD at PC1	σ_2	0.0133	0.0156	0.0156
19	SD at PC2	σ_3	0.0109	0.0123	0.0123
20	SD at PC3	σ_4	0.0106	0.0149	0.0149
21	SD at PC4	σ_5	0.0122	0.0175	0.0175
22	SD at PC5	σ_6	0.0099	0.0132	0.0132
23	SD at class (hardcore game)	σ_7	0.0084	0.0108	0.0108
24	SD at class (tools)	σ_8	0.0119	0.0158	0.0158
25	SD at class (shopping)	σ_9	0.009	0.0111	0.0111
26	SD at class (information)	σ_{10}	0.0104	0.0109	0.0109

of the semantic vectors with respect to heterogeneity in the marginal utility of usage.

8 Defining Relevant Markets

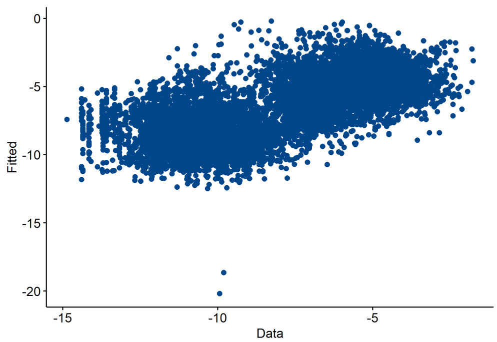
In the antitrust policy, the relevant market of a product typically needs to be defined to initiate the investigation of the case. The definition can be based on qualitative information such as the product category, such as game app, music app, and chat app in case of the App industry, or can be based on the quantitative analysis of price and quantity. SSNIP test is one of such methodology to define a relevant market. However, this test is not directly applicable to a free product because it checks whether the hypothetical monopolist owning the product can profitably increase the price when it owns other products. Without prices, whether the price increases cannot be determined.

Table 7: Estimation results

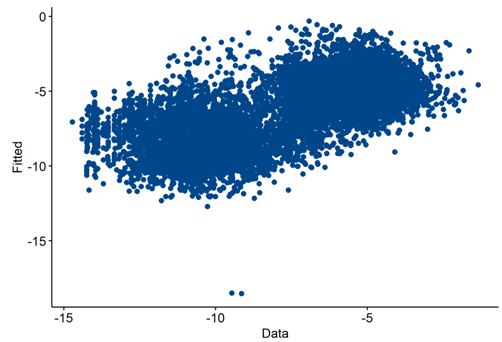
	Description	Parameter	Low-dimensional		High-dimensional
			Homogeneous	Heterogeneous	Heterogeneous
Download					
1	Intercept	β_{d1}	-9.260	-9.244	-24.908
2	PC1	β_{d2}	-1.997	-1.689	-3.600
3	PC2	β_{d3}	-0.260	-0.159	3.785
4	PC3	β_{d4}	4.381	4.137	2.118
5	PC4	β_{d5}	-5.792	-5.961	-13.717
6	PC5	β_{d6}	1.400	0.668	1.233
7	Class (hardcore game)	β_{d7}	-0.049	-0.080	-0.171
8	Class (tools)	β_{d8}	2.677	2.595	2.228
9	Class (shopping)	β_{d9}	1.843	1.856	0.919
10	Class (information)	β_{d10}	1.369	1.339	0.861
Usage					
11	Intercept	β_{u1}	0.409	0.348	0.813
12	PC1	β_{u2}	-0.011	-0.072	-0.446
13	PC2	β_{u3}	-0.059	-0.075	-0.276
14	PC3	β_{u4}	0.210	0.191	0.082
15	PC4	β_{u5}	-0.382	-0.370	-0.249
16	PC5	β_{u6}	0.343	0.494	0.356
17	Class (hardcore game)	β_{u7}	0.102	0.121	0.113
18	Class (tools)	β_{u8}	-0.023	-0.014	0.001
19	Class (shopping)	β_{u9}	-0.094	-0.104	-0.015
20	Class (information)	β_{u10}	-0.034	-0.038	-0.034

Table 8: Goodness-of-fits to download share and usage

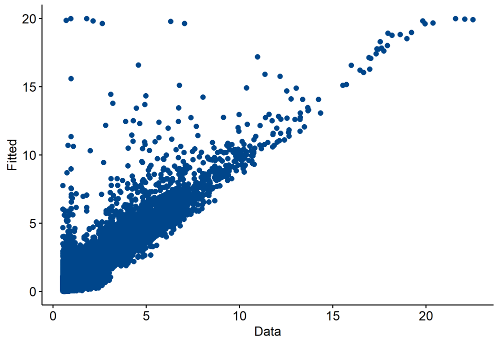
	In-sample		Out-of-sample	
	Download	Usage	Download	Usage
Residual standard deviation	2.1875	0.9934	2.1790	0.9404
Multiple R-squared	0.9311	0.8964	0.9309	0.9051
Adjusted R-squared	0.9311	0.8964	0.9309	0.9051



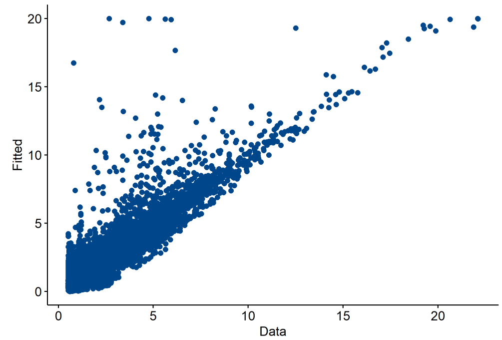
(a) In-sample: $\ln(s_{jt}/s_{0t})$



(b) Out-of-sample: $\ln(s_{jt}/s_{0t})$

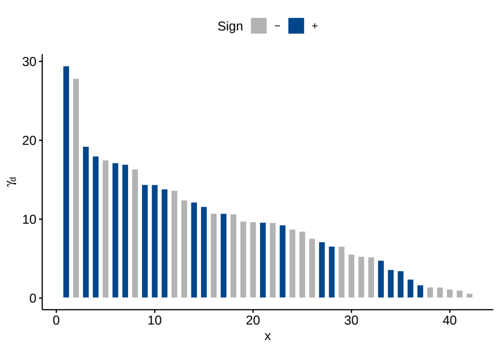


(c) In-sample: q_{jt}

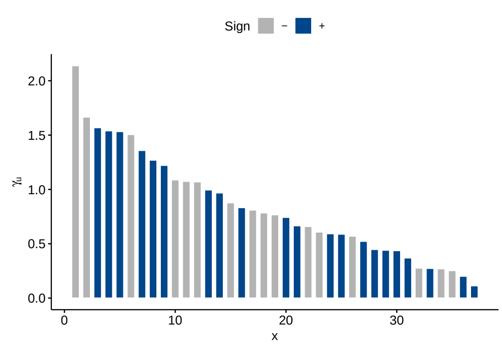


(d) Out-of-sample: q_{jt}

Figure 6: Fit to download share and usage



(a) γ_d from high-dimensional model



(b) γ_u from high-dimensional model

Figure 7: Estimation results

Even for a paid product, if the product generates revenue by showing advertisements, we need to consider the changes in advertising intensity to define a relevant market. Otherwise, the resulting market can be misleading.

Newman (2015) introduced the concept of *A Small, Non-transitory but Significant Increase in Cost (SSNIC)* test to resolve this problem. The SSNIC examines whether the cost to a consumer—not only the price—can be profitably increased. For the App economy, advertising intensity is the consumer's non-price cost that increases consumer's inconvenience while generates revenues for the producer. Thus, for free products, we can still define a relevant market, by focusing on advertising intensity. We call this version of the test a SSNIP test with both download price and advertising intensity.

In contrast to the standard demand model, the model in this paper endogenizes advertising intensity, thus, allowing us to define a relevant market in the App economy. This section discusses how to define a relevant market using the model and the estimation results from the previous section. It also illustrates the misleading nature of the definition of a relevant market using the standard demand model that endogenizes advertising intensity.

8.1 Methods

Store category As an example of relevant market definition using the existing product category, we use the categories defined in Google Play. Google Play defines Apps and Games as the upper-level categories. Below the Apps and Games categories, 49 categories are defined, as noted in Section 3.2.

Versions of small but significant and non-transitory increase in price (SSNIP) tests We introduce the SSNIP test that only uses download prices. We adopt the formalization introduced by Ivaldi and Lorincz (2011) for the SSNIP test, which attempts to describe the European Commission guidelines (European Commission, 1997) and U.S. guidelines for 1992.

Let $a = \{a_j\}_{j \in \mathcal{J}}$ and $F = \{F_j\}_{j \in \mathcal{J}}$ be the advertising intensity and download prices of the apps and $s(a, F) = \{s_j(a, F)\}_{j \in \mathcal{J}}$, $q(q, F) = \{q_j(a, F)\}_{j \in \mathcal{J}}$, and $\pi(a, F) = \{\pi_j(a, F)\}_{j \in \mathcal{J}}$ be the equilibrium download shares, usages, and profits under (a, F), respectively. Let (a^*, F^*) denote the benchmark equilibrium.

Then, the SSNIP relevant market of app j is formally defined as follows: Let $\mathcal{M} \subset \mathcal{J}$ and $j \in \mathcal{M}$. Let F^{SSNIP} be a download price equal to $(1 + \kappa)F_l^*$ if $l \in \mathcal{M}$, and F_l^*, where $0 < \kappa \leq 0.1$. Then, $\mathcal{M}$ is the SSNIP relevant market of app j if and only if:

1. $\Delta\pi_{\mathcal{M}}^{SSNIP} > 0$, where

$$\Delta\pi_{\mathcal{M}}^{SSNIP} \equiv \left(\frac{\sum_{l \in \mathcal{M}} [\pi_l(a^*, F^{SSNIP}) - \pi_l(a^*, F^*)]}{\sum_{l \in \mathcal{M}} \pi_l(a^*, F^*)} \right); \tag{50}$$

2. for all $\mathcal{M}' \subset \mathcal{J}$ such that $j \in \mathcal{M}'$ and $\mathcal{M}'$ satisfies (1), $\#(\mathcal{M}) \leq \#(\mathcal{M}')$.

Note that the definition of the relevant market only refers to the changes in the download prices and the advertising intensity is fixed when calculating the change in profits using equation (50). Thus, we cannot apply the test for apps without positive download prices. The resulting relevant market definitions can also be misleading because it does not take into account profit changes attributable to advertising changes. Moreover, the estimated demand function is mis-specified without endogenous advertising intensity: it further biases the relevant market definition. In a version of the SSNIP test with both download prices and advertising, a^* is also increased by 5% to a^{SSNIP}.

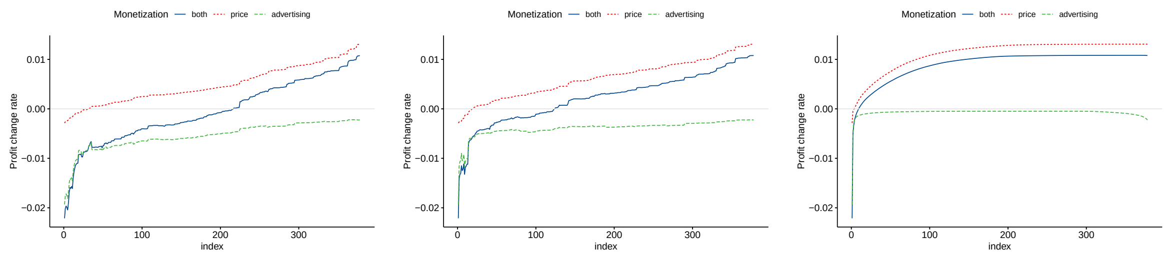
Order of testing price/cost increase For SSNIP or SSNIC tests, we sequentially add new products to the portfolio of the hypothetical monopolist. The profit change attributes to SSNIP or SSNIC depends on the order of adding products. This definition of a relevant market only refers to the minimal set of products that increases profits, but is silent about the procedure to determine the order to achieve the minimal set. In practice, the analyst often picks up an ad hoc but intuitive criterion such as the degree of the cross-price elasticity to determine the order of adding products. Following this practice, we consider an order based on the size of cross price elasticity with the target app and one based on the similarity in the semantic vector space measured by the cosine similarity. Moreover, we use a greedy strategy, in which we sequentially pick up an app from the remaining apps that maximizes the change in profits using SSNIP.

8.2 Comparing category, SSNIP, and SSNIC based relevant markets

Illustration with a news app Because of the confidentiality term of the data contract, we can only pick up apps based on the existing product category rankings, and must anonymize the app names. First, for illustration, we show the results for a top news app with the largest number of downloads in the category. We pick up a news app for illustration, because the literature on ad-sponsored media often focused on news media, such as newspapers and cable TV.

Figure 8 plots the change in profits of the hypothetical monopolist along the path of the SSNIP tests. In the plots, the x-axis indicates the index of added apps and the y-axis indicates the change in profits when the price and/or advertising intensity is increased by 5%. The top left panel indicates the apps ordered by elasticity, the right top indicates when ordered by similarity, and the bottom panel indicates the greedy strategy. First, the figure indicates that the results of the SSNIP tests can be substantially different if one ignores either the download price or advertisements. For the news app, the SSNIP test only with price tends to find a too small relevant market, whereas the SSNIP test only with advertising tends to find a too large relevant market. Second, although often dismissed, the relevant market definition is significantly sensitive to the order of the apps used to conduct the SSNIP tests.

Figure 9 demonstrates how a relevant market based on the SSNIP test can differ from a manually chosen product category. It is a tree map, in which the size of the rectangles represents the number of downloads. The top left panel is from similarity-based ordering, the top right panel is from



(a) Ordered by elasticity (b) Ordered by similarity (c) Ordered by greedy

Figure 8: SSNIP test: News & Magazines top app

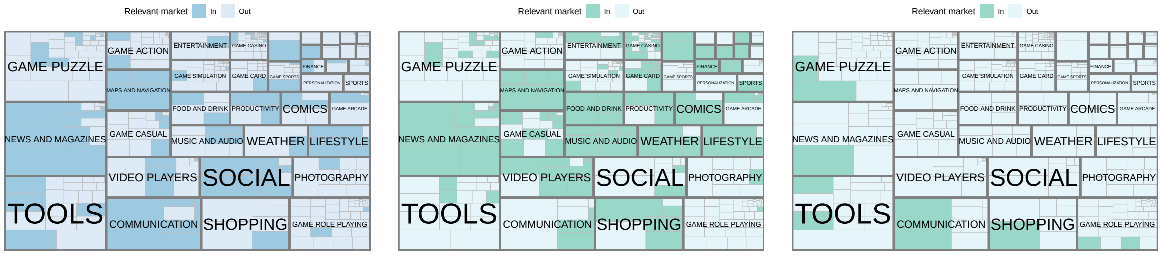
elasticity-based ordering, and the bottom panel is from greedy-based ordering. The rectangles of the apps included in the SSNIP-based relevant market are colored dark. Apps in the News & Magazines category are mostly included in the relevant market., which validates our procedure. However, we observe that the product category can be misleading, because the figure indicates that apps that belong to other categories, such as weather, maps and navigation, and lifestyle, can constitute the same market from a consumer's perspective. Table 9 shows the number of apps, the share of the target app, and the HHI of each market based on the product category and the SSNIP test. According to the relevant market based on the greedy strategy, the market share of the top app, and the HHI of the relevant market are higher than those based on the product category. Nevertheless, the market appears competitive enough for this top news app.

Relevant markets for top apps in each category Next, we select apps from each product category with the largest number of downloads. Then, we defined relevant markets by SSNIP for both download prices and advertisements using the greedy strategy. The results are summarized in Table 10. The number of apps that comprise a relevant market is smaller than the number of apps in each category. However, this does not mean that the market share of the top app and the concentration of the relevant market are always higher under the SSNIP test than under the category-based market. For example, for the top adventure game app, the number of apps is four with a SSNIP test compared with 15 in the category. Nevertheless, the share and HHI are lower in the relevant market with a SSNIP test than in the category market. This result implies that the arbitrary definition of a relevant market can lead to a misleading outcome.

9 Merger Simulation

9.1 Split Simulation of Major Communication Apps

Our framework enables us to conduct a full-equilibrium merger simulation. First, for illustration, we conduct a split simulation of a major communication app from a parent company that owns another major communication app. Because of confidentiality, we cannot disclose the name of the



(a) Ordered by similarity (b) Ordered by elasticity (c) Ordered by greedy

Figure 9: Relevant market by SSNIP test with download share: News & Magazines top app

Table 9: Comparison of share and HHI of News & Magazines top app between SSNIP and category based relevant markets

		Number		Share		HHI	
	Order	SSNIP	Category	SSNIP	Category	SSNIP	Category
1	By similarity	123	18	0.057	0.251	0.028	0.154
2	By elasticity	215	18	0.051	0.251	0.017	0.154
3	By greedy	9	18	0.181	0.251	0.169	0.154

app and the developer. Table 11 summarize the change in the key variables and welfare measures. By splitting this major app from the parent company, the download prices and advertisements substantially decrease among all parties. This results in a 2.2% and 0.4% increase in per-app profits for a spin-off app and outsiders, and a 0.3% decrine in per-app profits for the parent company. The consumer surplus increases for every app: 2.7% for a spin-off app, 0.6% for parent company apps, and 0.8% for outsider apps. The revenue of the platform—the Google Store in the current context—changes proportionally with the changes in the apps' profits. In total, the social surplus increases for the spin-off app by 2.5%, 0.1% for the parent company apps, and 0.6% for the outsider apps.

9.2 Merger Simulation of Top Apps

Next, to further evaluate the competitiveness of the app market, we again pick up top apps from each product category in terms number of downloads. We consider a situation in which the developer of each top app acquires the second to fifth closest apps in terms of elasticity. Figure 10 summarizes the change in prices, advertisements, and downloads after the merger in each product category, and 11 summarizes the effects on welfare measures. The magnitude of the effects differ across apps; however, in general, download prices increase the most among acquired apps, whereas advertisements increase the most among acquiring apps. As a result, both consumer and total surpluses decline from 0.5 to a few percent for each app. One of the interesting patterns that emerges from this analysis is that the increase in the price of acquired apps often comes at the cost of lower profits of the acquired apps. Panels (c) and (d) of Figure 11 show that the profits of acquired apps

Table 10: Comparison of share and HHI between SSNIP and category based relevant markets: Top apps

		Number		Share		HHI	
	Top app in:	SSNIP	Category	SSNIP	Category	SSNIP	Category
Game							
1	Action	4	31	0.23	0.14	0.302	0.082
2	Adventure	4	15	0.088	0.533	0.365	0.42
3	Arcade	7	11	0.205	0.672	0.198	0.519
4	Board	3	5	0.144	0.83	0.406	0.7
5	Card	5	17	0.065	0.272	0.438	0.148
6	Casino	3	69	0.194	0.188	0.459	0.062
7	Casual	4	20	0.266	0.14	0.298	0.093
8	Educational	2	3	0.239	0.797	0.636	0.673
9	Puzzle	4	47	0.447	0.187	0.318	0.076
10	Racing	3	3	0.115	0.492	0.602	0.495
11	Role Playing	11	57	0.015	0.037	0.153	0.041
12	Simulation	1	38	1	0.094	1	0.063
13	Sports	2	12	0.387	0.413	0.525	0.237
14	Strategy	3	5	0.135	0.694	0.593	0.54
Application							
15	Beauty	5	1	0.146	1	0.252	1
16	Books and Reference	2	2	0.308	0.996	0.573	0.992
17	Business	2	2	0.277	0.997	0.599	0.994
18	Comics	4	6	0.258	0.249	0.298	0.213
19	Communication	3	9	0.77	0.606	0.625	0.404
20	Education	3	11	0.15	0.708	0.491	0.559
21	Entertainment	4	11	0.269	0.375	0.302	0.252
22	Finance	2	2	0.215	0.504	0.662	0.5
23	Food and Drink	5	5	0.126	0.341	0.372	0.229
24	Health and Fitness	4	6	0.171	0.722	0.318	0.582
25	Lifestyle	4	9	0.428	0.369	0.349	0.233
26	Maps and Navigation	4	7	0.467	0.385	0.338	0.257
27	Music and Audio	5	11	0.136	0.317	0.366	0.173
28	News and Magazines	9	18	0.181	0.251	0.169	0.154
29	Personalization	7	8	0.107	0.982	0.252	0.964
30	Photography	4	14	0.319	0.164	0.306	0.115
31	Productivity	4	9	0.269	0.363	0.279	0.265
32	Shopping	4	7	0.271	0.315	0.406	0.213
33	Social	6	11	0.194	0.358	0.278	0.239
34	Sports	4	2	0.204	0.596	0.308	0.519
35	Tools	4	23	0.467	0.134	0.338	0.083
36	Travel and Local	4	2	0.375	0.706	0.305	0.585
37	Video Players	8	18	0.108	0.241	0.211	0.153
38	Weather	4	4	0.554	0.704	0.391	0.544

Table 11: Effects of spinning out a major app

	Outsider	Parent	Spin-off
Price (JPY)	-27.1	-90.7	-43.8
Advertising (Count/Hour)	-17.1	-619.6	-0.3
Download (Count/Week)	28.5	45.5	816.7
App profit (%)	0.4	-0.3	2.2
Consumer surplus (%)	0.8	0.6	2.7
Platform profit (%)	0.4	-0.3	2.2
Total surplus (%)	0.6	0.1	2.5

Table 12: Effects of platform fee reduction to zero

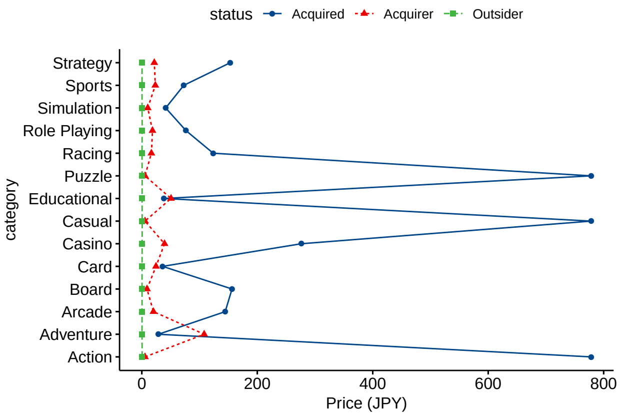
Price (JPY)	Advertising (Count/Hour)	Download (Count/Week)
-72.9	-232.8	101.6

App profit (%)	Consumer surplus (%)	Platform Profit (%)	Total surplus (%)
51.5	10.6	-100.0	8.4

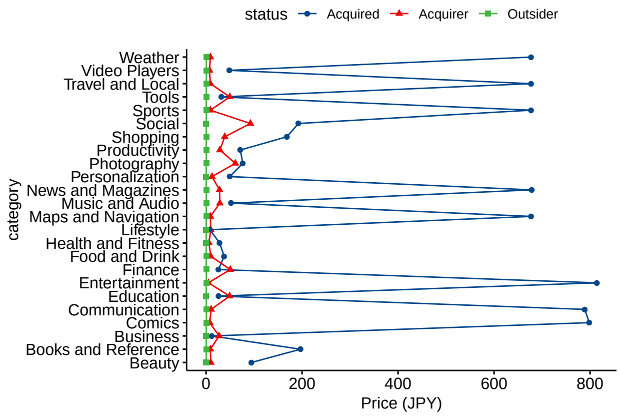
decline when their download prices were increased, which by construction increases the total profits of the merged company. Thus, after mergers, the acquiring developer finds it profitable to "kill" the acquired apps that are close substitutes to the top app.

10 Platform Fee Reduction

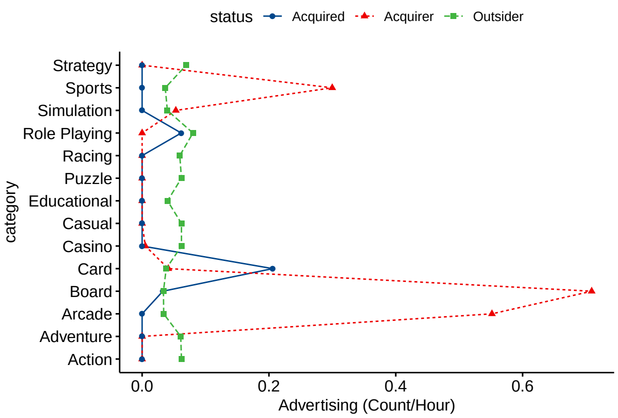
The framework can be used to conduct other type of counterfactual analysis. For example, we can ask what the effects of platform fee reduction would be. Currently, Google Play charges a 30% platform fee based on the download and in-app purchase revenue. Table 12 provides the changes in the key variables and welfare measures when the platform fee is reduced to zero and indicates that the download price declines by JPY 72.9. Moreover, advertisements substantially decline. Two mechanisms worked behind this change. First, the price declines because the double marginalization was removed. Second, the price increased and advertisements decreased because the marginal return of the increasing download price increases. The result is a 51.5% increase in app profits and a 10.6% increase in the consumer surplus. In total, the surplus increases by 8.4%.



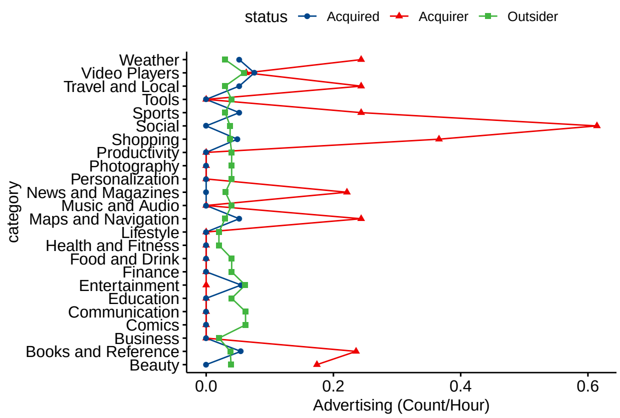
(a) Game: Price



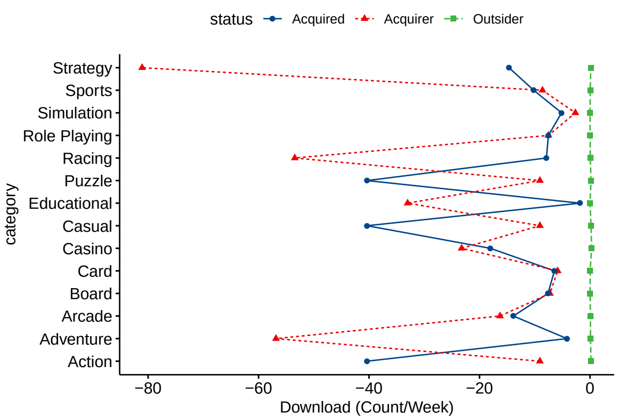
(b) Application: Price



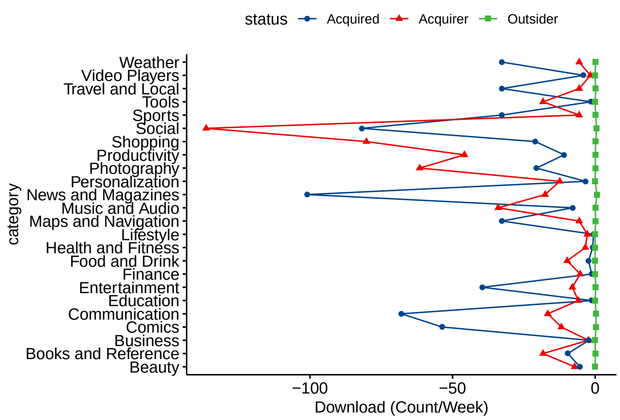
(c) Game: Advertising



(d) Application: Advertising

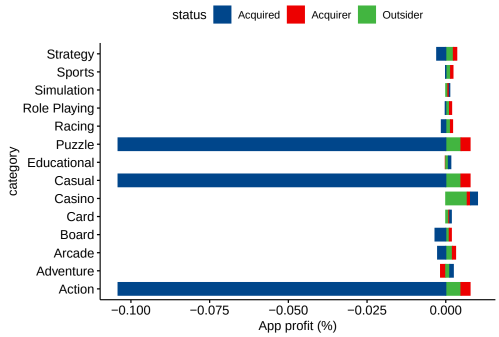


(e) Game: Download

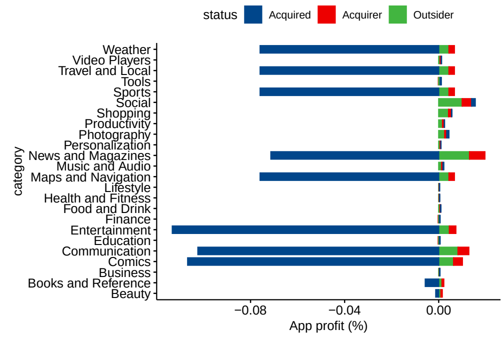


(f) Application: Download

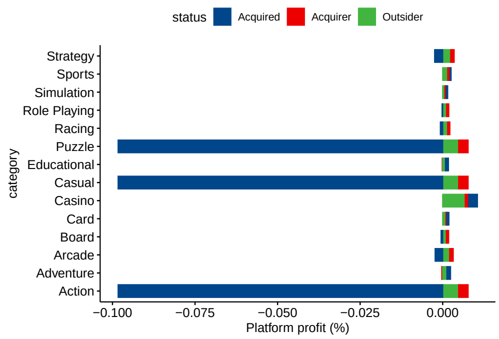
Figure 10: Effects of mergers among top 5 apps



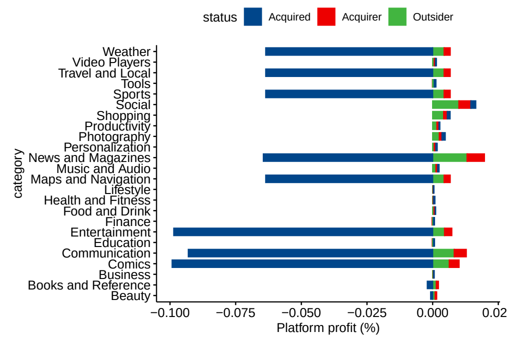
(a) Game: App profit



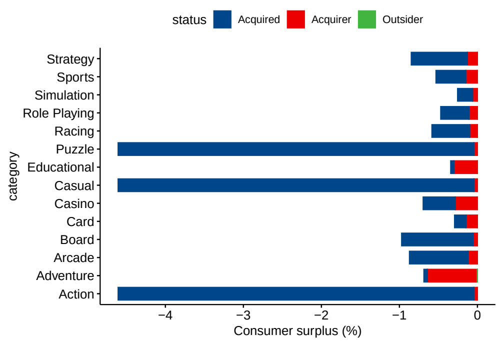
(b) Application: App profit



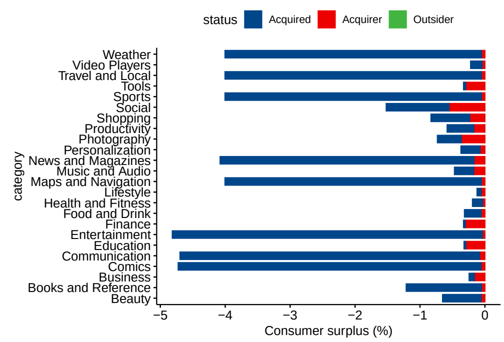
(c) Game: Platform profit



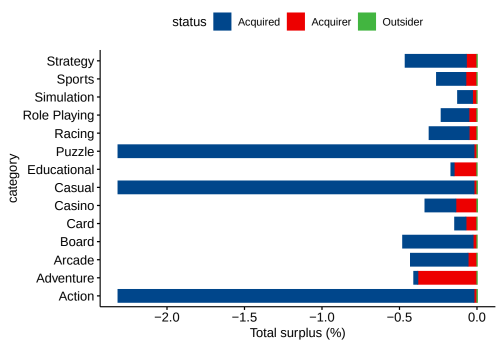
(d) Application: Platform profit



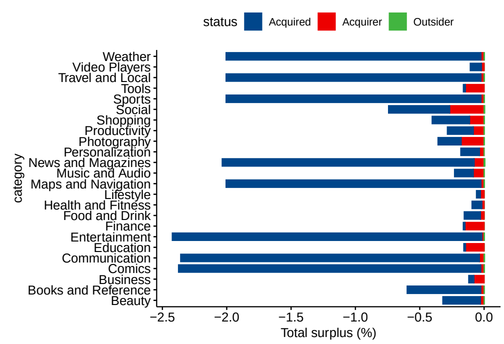
(e) Game: Consumer surplus



(f) Application: Consumer surplus



(g) Game: Total surplus



(h) Application: Total surplus

Figure 11: Effects of mergers among top 5 apps

11 Conclusion

In this paper, we proposed a new model of imperfect competition of ad-sponsored media and applied it to the mobile app industry in Japan. The model allowed firms to compete with respect to both pricing and advertising revenues and endogenously select business models (paid or free media). The model incorporated word embeddings to numerically represent product information, allowing relevant markets to be defined in newly created and quickly redefined industries. We estimated the model using a rich data set on consumer downloads, usage, in-app purchases, and price and advertising information. The model introduced a novel identification strategy of unobserved advertising intensity that exploits a unique condition in the mobile app industry: that no direct marginal cost is associated with gathering sponsored advertisements. Because of these newly introduced features of the model, we could conduct a version of the SSNIP test in which both price and sponsored advertisements are increased. Our analysis showed that a SSNIP test that ignores either price or sponsored advertisements is misleading in a merger analysis. It warns an antitrust authority to carefully define relevant markets in this industry. Our merger analysis showed that, in some cases, acquiring companies find it profitable to "kill" acquired apps by substantially increasing prices, to maximize the company's total profits. Finally, we studied the effect of reducing the platform fee to zero, which showed that the total surplus increases by 8.4% through the removal of the double marginalization between the platform and developers and a reduction in the distortion from pricing to advertising.

This paper has several limitations. First, we assume that all firms use the ad network as price takers. In reality, some developers would not use the ad network to exert their market power in the ad market. To address this issue, we must directly observe individual advertising intensity and advertising revenue. Second, the coefficients in the usage-related indirect utility were deterministic. Making them random increases the difficulty of the computation, but would be desirable. Third, the market definition is restricted to the mobile app market. From a consumer's perspective, some apps can be a substitute for a service outside the mobile app market. For example, mobile payment services competess with credit cards. Studying the interactions between the mobile app market and the outside market is essential in analyzing the app economy.

References

Ackerberg, Daniel A. and Marc Rysman, "Unobserved Product Differentiation in Discrete-Choice Models: Estimating Price Elasticities and Welfare Effects," *The RAND Journal of Economics*, 2005, *36* (4), 771–788.

Affeldt, Pauline, Lapo Filistrucchi, and Tobias J. Klein, "Upward Pricing Pressure in Two-sided Markets," *The Economic Journal*, November 2013, *123* (572), F505–F523.

Allcott, Hunt, Luca Braghieri, Sarah Eichmeyer, and Matthew Gentzkow, "The Welfare Effects of Social Media," *American Economic Review*, March 2020, *110* (3), 629–676.

Anderson, Simon P. and Jean J. Gabszewicz, "Chapter 18 The Media and Advertising: A Tale of Two-Sided Markets," in Victor A. Ginsburg and David Throsby, eds., *Victor A. Ginsburg and David Throsby, eds.*, Vol. 1 of *Handbook of the Economics of Art and Culture*, Elsevier, 2006, pp. 567–614.

_ , Øystein Foros, and Hans Jarle Kind, "The Importance of Consumer Multihoming (Joint Purchases) for Market Performance: Mergers and Entry in Media Markets," *Journal of Economics & Management Strategy*, 2019, *28* (1), 125–137.

Angus, Ryan W., "Problemistic Search Distance and Entrepreneurial Performance," *Strategic Management Journal*, December 2019, *40* (12), 2011–2023.

App Annie, "App Annie 2017 Retrospective Report," Technical Report 2017.

_ , "The State of Mobile 2019," Technical Report 2019.

Asahara, Masayuki, "NWJC2Vec: Word Embedding Dataset from 'NINJAL Web Japanese Corpus'," *Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication*, 2018, *24* (2), 7–25.

_ , Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi, "Archiving and Analysing Techniques of the Ultra-Large-Scale Web-Based Corpus Project of NINJAL, Japan," *Alexandria*, 2014, *25* (1-2), 129–148.

Barlow, Matthew A., J. Cameron Verhaal, and Ryan W. Angus, "Optimal Distinctiveness, Strategic Categorization, and Product Market Entry on the Google Play App Platform," *Strategic Management Journal*, April 2019, pp. 1219–1242.

Belloni, Alexandre and Victor Chernozhukov, "Least Squares after Model Selection in High-Dimensional Sparse Models," *Bernoulli*, May 2013, *19* (2), 521–547.

Berry, Steven and Ariel Pakes, "The Pure Characteristics Demand Model," *International Economic Review*, December 2007, *48* (4), 1193–1225.

_ , James Levinsohn, and Ariel Pakes, "Automobile Prices in Market Equilibrium," *Econometrica*, July 1995, *63* (4), 841.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, December 2017, *5*, 135–146.

Bordalo, Pedro, Nicola Gennaioli, and Andrei Shleifer, "Competition for Attention," *The Review of Economic Studies*, November 2015, *83* (2), 481–513.

Brynjolfsson, Erik and JooHee Oh, "The Attention Economy: Measuring the Value of Free Digital Services on the Internet," in "International Conference on Information Systems," Vol. 4 December 2012, pp. 3243–3261.

_ , Avinash Collis, and Felix Eggers, "Using Massive Online Choice Experiments to Measure Changes in Well-Being," *Proceedings of the National Academy of Sciences*, April 2019, *116* (15), 7250–7255.

Calvano, Emilio and Michele Polo, "Strategic Differentiation by Business Models: Free-To-Air and Pay-TV," *The Economic Journal*, July 2019, *130* (625), 50–64.

Carare, Octavian, "The Impact of Bestseller Rank on Demand: Evidence from the App Market," *International Economic Review*, 2012, *53* (3), 717–742.

Cayseele, Patrick Van and Stijn Vanormelingen, "Merger Analysis in Two-Sided Markets: The Belgian Newspaper Industry," *Review of Industrial Organization*, May 2019, *54* (3), 509–541.

comScore, "The Global Mobile Report," 2017.

Crawford, Gregory S., Robin S. Lee, Michael D. Whinston, and Ali Yurukoglu, "The Welfare Effects of Vertical Integration in Multichannel Television Markets," *Econometrica*, 2018, *86* (3), 891–954.

Crémer, Jacques, Yves-Alexandre de Montjoye, Heike Schweitzer, European Commission, and Directorate-General for Competition, *Competition Policy for the Digital Era.* May 2019. OCLC: 1111125847.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, 1990, *41* (6), 391–407.

Deng, Yiting, Anja Lambrecht, and Yongdong Liu, "Spillover Effects and Freemium Strategy in Mobile App Market," *SSRN Electronic Journal*, 2018.

Eisenstein, Jacob., *Introduction to Natural Language Processing*, Cambridge, MA: The MIT Press, 2019.

Eizenberg, Alon, "Upstream Innovation and Product Variety in the U.S. Home PC Market," *The Review of Economic Studies*, July 2014, *81* (3), 1003–1045.

Emch, Eric and T Scott Thompson, "Market Definition and Market Power in Payment Card Networks," *Review of Network Economics*, 2006, *5* (1), 45–60.

Ershov, Daniel, "Consumer Product Discovery Costs, Entry, Quality and Congestion in Online Markets," 2020.

European Commission, *Commission Notice on the Definition of Relevant Market for the Purposes of Community Competition Law*, Brussels: European Commission, 1997.

Evans, D. S. and M. D. Noel, "The Analysis of Mergers That Involve Multisided Platform Businesses," *Journal of Competition Law and Economics*, September 2008, *4* (3), 663–695.

Fan, Ying, "Ownership Consolidation and Product Characteristics: A Study of the US Daily Newspaper Market," *American Economic Review*, 2013, *103* (5), 1598–1628.

Filistrucchi, Lapo, Tobias J. Klein, and Thomas O. Michielsen, "Assessing Unilateral Merger Effects in a Two-Sided Market: An Application to the Dutch Daily Newspaper Market," *Journal of Competition Law and Economics*, 2012, *8* (2), 297–329.

Firth, J. R., *Papers in Linguistics, 1934-1951*, London: Oxford University Press, 1957.

Gandhi, Amit and Jean-François Houde, "Measuring Substitution Patterns in Differentiated Products Industries," Working Paper 26375, National Bureau of Economic Research October 2019. Series: Working Paper Series.

Gentzkow, Matthew and Jesse M. Shapiro, "What Drives Media Slant? Evidence From U.S. Daily Newspapers," *Econometrica*, 2010, *78* (1), 35–71.

_ , Bryan Kelly, and Matt Taddy, "Text as Data," *Journal of Economic Literature*, September 2019, *57* (3), 535–574.

_ , Jesse M. Shapiro, and Michael Sinkinson, "Competition and Ideological Diversity: Historical Evidence from US Newspapers," *American Economic Review*, October 2014, *104* (10), 3073–3114.

Ghose, Anindya and Sang Han, "Estimating Demand for Mobile Applications in the New Economy," *Management Science*, 2014, *60* (6), 1470–1488.

Goolsbee, Austan and Peter J Klenow, "Valuing Consumer Products by the Time Spent Using Them: An Application to the Internet," *American Economic Review*, 2006, *96* (2), 7.

Grigolon, Laura and Frank Verboven, "Nested Logit or Random Coefficients Logit? A Comparison of Alternative Discrete Choice Models of Product Differentiation," *Review of Economics and Statistics*, December 2014, *96* (5), 916–935.

Han, Sang Pil, Sungho Park, and Wonseok Oh, "Mobile App Analytics: A Multiple Discrete-Continuous Choice Framework," *MIS Quarterly*, 2016, *40* (4), 983–1008.

Hausman, Jerry A., "Valuation of New Goods under Perfect and Imperfect Competition," in "The Economics of New Goods," University of Chicago Press, January 1996, pp. 207–248.

Hoberg, Gerard and Gordon Phillips, "Text-Based Network Industries and Endogenous Product Differentiation," *Journal of Political Economy*, October 2016, *124* (5), 1423–1465.

Ifrach, Bar and Ramesh Johari, "The Impact of Visibility on Demand in the Market for Mobile Apps," *SSRN Electronic Journal*, 2014.

Ivaldi, Mark and Szabolcs Lorincz, "Implementing Relevant Market Tests in Antitrust Policy: Application to Computer Servers," *Review of Law and Economics*, 2011, *7* (1), 31–73.

Japan Fair Trade Commission, "The Revised Guidelines to Application of the Antimonopoly Act Concerning Review of Business Combination," December 2019.

Jara-Díaz, Sergio and Jorge Rosales-Salas, "Beyond Transport Time: A Review of Time Use Modeling," *Transportation Research Part A: Policy and Practice*, March 2017, *97*, 209–230.

Jeon, Doh-Shin, Byung-Cheol Kim, and Domenico Menicucci, "Second-Degree Price Discrimination by a Two-Sided Monopoly Platform," *Working Paper*, 2016.

Jeziorski, PrzemysŁaw, "Effects of Mergers in Two-Sided Markets: The US Radio Industry," *American Economic Journal: Microeconomics*, 2014, *6* (4), 35–73.

Kesler, Reinhold, Michael E. Kummer, and Patrick Schulte, "Mobile Applications and Access to Private Data: The Supply Side of the Android Ecosystem," *SSRN Electronic Journal*, 2017.

Kudo, Taku, "MeCab : Yet Another Part-of-Speech and Morphological Analyzer," 2005.

Kwark, Young and Paul A Pavlou, "On the Spillover Effects of Online Product Reviews on Purchases : Evidence from Clickstream Data," *SSRN Electronic Journal*, 2019.

Leyden, Benjamin T, "There ' s an App for That," 2018.

Lin, Song, "Two-Sided Price Discrimination by Media Platforms," *Marketing Science*, 2020, *39* (2), 317–338.

Liu, Yongdong, "Mobile App Platform Choice," *SSRN Electronic Journal*, 2017, pp. 1–48.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds., *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., 2013, pp. 3111–3119.

_ , Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, September 2013.

Morozov, Ilya, "Measuring Benefits from New Products in Markets with Information Frictions," *SSRN Electronic Journal*, 2019.

Newman, John M., "Antitrust in Zero-Price Markets: Foundations," *University of Pennsylvania Law Review*, 2015, *164*, 149–206.

P., Simon Anderson and Martin Peitz, "Media Sea-Saws: Winners and Losers in Platform Markets," *Journal of Economic Theory*, 2020, *186.*

Pantea, Smaranda and Bertin Martens, "The Value of the Internet as Entertainment in Five European Countries," *Journal of Media Economics*, January 2016, *29* (1), 16–30.

Pennington, Jeffrey, Richard Socher, and Christopher Manning, "Glove: Global Vectors for Word Representation," in "Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)" Association for Computational Linguistics Doha, Qatar 2014, pp. 1532–1543.

Pervin, Nargis, Narayan Ramasubbu, and Kaushik Dutta, "Habitat Traps in Mobile Platform Ecosystems," *Production and Operations Management*, October 2019, *28* (10), 2594–2608.

Petrin, Amil, "Quantifying the Benefits of New Products: The Case of the Minivan," *Journal of Political Economy*, 2002, *110* (4), 705–729.

Recode, "Spotify Says Apple Won't Approve a New Version of Its App Because It Doesn't Want Competition for Apple Music - Recode," https://www.recode.net/2016/6/30/12067578/spotify-apple-app-store-rejection June 2016.

Řehůřek, Radim and Petr Sojka, "Software Framework for Topic Modelling with Large Corpora," in "Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks" ELRA Valletta, Malta May 2010, pp. 45–50.

Sato, Susumu, "Freemium as Optimal Menu Pricing," *International Journal of Industrial Organization*, 2019, *63*, 480–510.

Sato, Toshinori, Taiichi Hashimoto, and Manabu Okumura, "Implementation of a Word Segmentation Dictionary Called Mecab-Ipadic-NEologd and Study on How to Use It Effectively for Information Retrieval (in Japanese)," in "Proceedings of the Twenty-Three Annual Meeting of the Association for Natural Language Processing" 2017, pp. NLP2017–B6–1.

Song, Minjae, "Measuring Consumer Welfare in the CPU Market: An Application of the Pure-Characteristics Demand Model," *The RAND Journal of Economics*, June 2007, *38* (2), 429–446.

_ , "Estimating Platform Market Power in Two-Sided Markets with an Application to Magazine Advertising," *SSRN Electronic Journal*, 2011.

The Wall Street Journal, "Facebook Seeks EU Antitrust Review of WhatsApp Deal: Source — Reuters," May 2014.

Train, Kenneth E., *Discrete Choice Methods with Simulation.*, second ed., Cambridge: Cambridge University Press, 2009.

A Algorithm and Proof of Convergence

We iterate the best-response mapping to compute the least equilibrium. A function $\Psi : X \to X$ is an increasing function if $x \geq x'$ implies $\Psi(x) \geq \Psi(y)$. A strategy profile (a, F) is the least equilibrium if (i) (a, F) is an equilibrium strategy profile, and (ii) for any equilibrium (a', F) strategy profile, $(a', F) \geq (a, F)$ holds.

Suppose that $w_m = w$ for all $m \in \mathcal{M}$. Then, we have $q_j = q_{mj}$, $e_j = e_{mj}$, and $s_j = s_{mj}$ for all $m \in \mathcal{M}$. In this case, the first-order condition (23) and (24) can be simplified into

$$\frac{\partial \Pi_d}{\partial F_j} = (1-\rho)s_j + \sum_{k \in \mathcal{J}_d} \frac{\partial s_k}{\partial F_j}[(1-\rho)(F_k + e_k) + q_k a_j r] \leq 0, \tag{51}$$

with equality if $F_j > 0$ for each $j \in \mathcal{J}_d$, and:

$$\frac{\partial \Pi_d}{\partial a_j} = s_j q_j r + s_j \frac{\partial q_j}{\partial a_j} a_j r + s_j(1-\rho)\frac{\partial e_j}{\partial a_j} + \sum_{k \in \mathcal{J}_d} \frac{\partial s_k}{\partial a_j}\{(1-\rho)(F_k + e_k) + q_k a_k r\} \leq 0, \tag{52}$$

To guarantee the convergence of the algorithm, we first show that the best-response mapping $\Psi \colon \mathbb{R}_+^{2J} \to \mathbb{R}_+^{2J}$ is an increasing function. Then, we show that the algorithm converges to the least equilibrium.

First, we show that the best-response mapping is an increasing function. Take any vector $\{a, F\} = \{a_j, F_j\}_{j \in \mathcal{J}}$. Let $\Psi(a, F) \coloneqq \{\Psi_d(a, F)\}_{d \in \mathcal{D}}$ be the profile best-response of app developer d, where $\Psi_d(a, F) = \{\Psi_{dj}(a, F)\}_{j \in \mathcal{J}_d}$, and $\Psi_{dj}(a, F) = \{\Psi^a_{dj}(a, F), \Psi^F_{dj}(a, F)\}$ is the advertising intensity and the download price of app j provided by developer d at the best-response. Take (a', F') such that $(a', F') \geq (a, F)$ and compare the values of $\Psi_{dj}(a, F)$ and $\Psi_{dj}(a', F')$. Several cases are classified:

1. If $\Psi_{dj}(a, F) >> 0$, then, by the first-order condition, a_j is invariant to the strategy of other firms, and F_j is increasing in the strategies of the other firms. Thus, $\Psi^a_{dj}(a', F') = \Psi^a_j(a, F)$ and $\Psi^F_{dj}(a', F') > \Psi^F_{dj}(a, F)$ hold.

2. If $\Psi^F_{dj}(a, F) = 0$ and $\Psi^a_{dj}(a, F) > 0$, then we can further classify two cases:

   - $\Psi^F_{dj}(a', F') = \Psi^F_{dj}(a, F) = 0$, $\Psi^a_{dj}(a', F') > 0$, and $\Psi^a_{dj}(a, F) > 0$, then, by the first-order conditions, a_j increases with the strategies of the other firms, and thus $\Psi^a_{dj}(a', F') > \Psi^a_{dj}(a, F)$ holds.
   - If $\Psi^F_{dj}(a', F') > 0$ and $\Psi^F_{dj}(a, F) = 0$, then, $\Psi^a_{dj}(a', F')$ and $\Psi^a_{dj}(a, F)$ satisfy

   $$q_j r + \frac{\partial q_j}{\partial a_j} a_j r + (1-\rho)\left(\frac{\partial e_j}{\partial a_j} - \frac{\alpha_{aj}}{\alpha_y} q_j\right) = 0$$

   at $a_j = \Psi^a_j(a', F')$, while

   $$q_j r + \frac{\partial q_j}{\partial a_j} a_j r + (1-\rho)\left(\frac{\partial e_j}{\partial a_j} - \frac{\alpha_{aj}}{\alpha_y} q_j\right) \geq 0$$

   at $a_j = \Psi^a_{dj}(a, F)$. Then, we observe that $\Psi^a_{dj}(a', F') > \Psi^a_{dj}(a, F)$ holds.

3. If $\Psi^F_{dj}(a,F) > 0$ and $\Psi^a_{dj}(a,F) = 0$, then $\Psi^a_{dj}(a',F') = 0$ as long as $\Psi^F_{dj}(a',F') > 0$ because both of $\Psi^F_{dj}(a,F) > 0$ and $\Psi^a_{dj}(a,F)$ hold only when

$$q_j r + \frac{\partial q_j}{\partial a_j} a_j r + (1-\rho)\left(\frac{\partial e_j}{\partial a_j} - \frac{\alpha_{aj}}{\alpha_y} q_j\right) \geq 0$$

holds, which does not depend on the strategy of other firms. Thus, by the first-order condition, $\Psi^F_{dj}(a',F') > \Psi^F_j(a,F)$, and thus $\Psi^a_{dj}(a',F') > \Psi^a_{dj}(a,F)$ holds.

4. If $\Psi_{dj}(a,F) = (0,0)$, then $\Psi_{dj}(a',F') \geq \Psi_{dj}(a,F)$ always holds because $(0,0)$ is a lower bound of the strategy space.

This holds for all $j \in \mathcal{J}_d$ and $d \in \mathcal{D}$. As a result, Ψ is an increasing function.

Next, we show the convergence of the algorithm. Let $(a^0, F^0) := (0,\ldots,0)$. For each m, let (a^{m+1}, F^{m+1}) be

$$(a^{m+1}_j, F^{m+1}_j) = \Psi_{dj}(a^m, F^m). \tag{53}$$

Define the operator Ψ^m as $\Psi^1(a,F) = \Psi(a,F)$, and $\Psi^{m+1}(a,F) = \Psi(\Psi^m(a,F))$ for $m = 2,\ldots$. This sequence is bounded above by the monopoly price and advertising intensity. Thus, it is an increasing sequence in a compact space (bounded above) and converges to some point (a,F) with property $\Psi(a,F) = (a,F)$. If $\Psi(a,F) \neq (a,F)$, then we have

$$\Psi(\lim_{m\to\infty}(a^m,F^m)) = \Psi((a,F)) \neq (a,F) = \lim_{m\to\infty}\Psi(a^m,F^m), \tag{54}$$

which contradicts the continuity of Ψ.

Finally, we show that this fixed point is the minimum fixed point. Take any fixed point (a',F'). Then, we have

$$(a,F) = \lim_{m\to\infty}\Psi^m(0) \leq \lim_{m\to\infty}\Psi^m(a',F') = (a',F'). \tag{55}$$

B Quadratic-Programming for Eliciting Advertising Intensities

In this section, we outline the quadratic-programming procedure for eliciting advertising intensities used in Section 7.

Fix θ. Given the average indirect utilities $(\delta_j)_{j\in\mathcal{J}}$, and parameters θ, we can compute $\partial s_{mk}/\partial a_j$ independent of $(a_j)_{j\in\mathcal{J}}$. Making use of this property, we derive the profile of advertising intensity by conducting the following quadratic-programming procedure. First, to elicit the advertising intensities $\{a_j\}_{j\in\mathcal{J}}$, we utilize the first-order conditions for profit-maximizing advertising intensities given each parameters, which is characterized by

$$\begin{aligned} &s_j\left(q_j r + \frac{\partial q_j}{a_j}(a_j r - \lambda_j) + (1-\rho)\frac{\partial e_j}{\partial a_j}\right) \\ &-\alpha_{aj} q_j \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial \delta_j}((1-\rho)(F_j + e_j) + q_j(a_j r - \lambda_j)) + \mu_j = 0, \mu_j a_j = 0, \mu_j \geq 0, a_j \geq 0, \end{aligned} \tag{56}$$

for all $j \in \mathcal{J}$. Given the data $(s_j, q_j, e_j)_{j\in\mathcal{J}}$, mean indirect utilities $(\delta_j)_{j\in\mathcal{J}}$, and parameters θ, we can compute the simulated value of $\partial s_k/\partial\delta_j$. Let

$$\begin{aligned}\omega_j(a) =& s_j\left(q_j r + \frac{\partial q_j}{\partial a_j}(a_j r - \lambda_j) + (1-\rho)\frac{\partial e_j}{\partial a_j}\right) - \alpha_{aj} q_j \sum_{k\in\mathcal{J}_d}\frac{\partial s_k}{\partial \delta_j}((1-\rho)(F_k + e_k) + q_k(a_k r - \lambda_k)) \\ =& s_j q_j r - s_j\frac{\partial q_j}{\partial a_j}\lambda_j + (1-\rho)\frac{\partial e_j}{\partial a_j} - \alpha_{aj} q_j \sum_{k\in\mathcal{J}_d}\frac{\partial s_k}{\partial \delta_j}((1-\rho)(F_k+e_k) - q_k\lambda_k) \\ &+ s_j\frac{\partial q_j}{\partial a_j} r a_j - \alpha_{aj} q_j \sum_{k\in\mathcal{J}_d}\frac{\partial s_k}{\partial \delta_j} q_k r a_k.\end{aligned} \tag{57}$$

Let $\tilde{a}_j = q_j a_j$ and $\tilde{a} := (\tilde{a}_j)_{j\in\mathcal{J}}$. Then, $\omega_j(a)$ can be written as a function of $\tilde{a}$, $\tilde{\omega}(\tilde{a})$ as

$$\tilde{\omega}_j(\tilde{a}) = s_j q_j r + (1-\rho)\frac{\partial e_j}{\partial a_j} - \alpha_{aj} q_j \sum_{k\in\mathcal{J}_d}\frac{\partial s_k}{\partial \delta_j}(1-\rho)(F_k+e_k) + s_j\frac{\partial q_j}{\partial a_j}\frac{r}{q_j}\tilde{a}_j - \alpha_{aj} q_j \sum_{k\in\mathcal{J}_d}\frac{\partial s_k}{\partial \delta_j} r\tilde{a}_k, \tag{58}$$

which can be written in vector form

$$\tilde{\omega}(\tilde{a}) = \gamma - \Gamma\tilde{a}, \tag{59}$$

where $\gamma = (\gamma_j)_{j\in\mathcal{J}}$ is given by

$$\gamma_j = s_j q_j r + (1-\rho)\frac{\partial e_j}{\partial a_j} - \alpha_{aj} q_j \sum_{k\in\mathcal{J}_d}\frac{\partial s_k}{\partial \delta_j}(1-\rho)(F_k+e_k)$$

for each $j \in \mathcal{J}_d$, and $\Gamma = (\Gamma_{ij})_{i,j\in\mathcal{J}}$ is given by

$$\Gamma_{ij} = \begin{cases} s_j\frac{\partial q_j}{\partial a_j}\frac{r}{q_j} - \alpha_{aj} q_j \frac{\partial s_j}{\partial \delta_j} r & \text{if } i = j, \\ -\alpha_{aj} q_j \frac{\partial s_j}{\partial \delta_j} r & \text{if } i \neq j, \text{ and there exists } d \in \mathcal{D} \text{ such that } i, j \in \mathcal{J}_d \\ 0 & \text{otherwise.}\end{cases}$$

Then, we claim that we can obtain the values of $\{a_j\}_{j\in\mathcal{J}}$ by solving the following minimization problem

$$\max_{\tilde{a}} \gamma'\tilde{a} - \frac{1}{2}\tilde{a}'\Gamma\tilde{a} \quad \text{subject to } \tilde{a} \geq 0 \tag{60}$$

and plugging the solution into $a_j = \tilde{a}_j/q_j$.

Whenever possible, solving this maximization problem yields exactly the same first-order condition with the original problem at the true parameters (θ) and average indirect utilities $(\delta_j)_{j\in\mathcal{J}_d}$ which is shown by the following logic. First, when the quadratic programming (60) has a concave objective function, we can find a unique pair of profiles of advertising intensities and Lagrange multipliers $(a_j, \mu_j)_{j\in\mathcal{J}}$ that achieves the global maximum of the objective function. Next, we claim that the true values $(\tilde{a}_j) = (q_j a_j)$ and $(\mu_j)_{j\in\mathcal{J}}$ in the equilibrium satisfy the necessary and sufficient condition for the solution to this quadratic programming. The Kuhn-Tucker condition for the quadratic programming (60) is given by

$$\tilde{\omega}(\tilde{a}) + \tilde{\mu}_j = 0, \tilde{\mu}_j\tilde{a}_j = 0, \tilde{\mu}_j \geq 0, \tilde{a}_j \geq 0,$$

which is equivalent to the first-order condition of the original model (56). Thus, the solution to this quadratic programming corresponds to the equilibrium value at the true parameter value.

C Monte Carlo Simulation

Tables 13-15 illustrate how the equilibrium prices and advertising intensities depend on the underlying model parameters. In the simplest case, both α_{aj} and η_j are common across the apps. In each environment, we set the baseline parameters as $\alpha_{aj} = 0.05$, $\alpha_y = 0.1$ $\eta_j = 2$, $\lambda_j = 0$ for all j, $\beta_u = 0.1$, $\beta_d = 1$, and $\sigma_k = 0$ for all k. For advertising prices and wages, we set $r = 5$ and $w = 0$, and for firm-level quality shocks, we set $\xi_{uj} = 0$, $\xi_{dj} = -3$, and $\xi_{ej} = 0.0001$ for all j. We consider three scenarios: symmetric duopoly, asymmetric duopoly with heterogeneity in ξ_{ej}, and asymmetric duopoly with heterogeneity in ownership structure. In an asymmetric duopoly with heterogeneity in the ownership structure, one firm provides three products, and another firm provides only one product.

Table 13 shows how equilibrium advertising intensities and download prices depend on parameters α_y, α_{aj}, and η_j. We set the value of the equilibrium variable under baseline parameters to 1 and show the relative values of the equilibrium variables with different parameters values. The greater the marginal utility of income, the greater the equilibrium advertising intensities and the smaller the equilibrium download prices. This occurs because of the substitution from download prices to advertising intensities to effectively collect revenues. A greater marginal utility of income increases consumers' costs for downloading apps, leading to lower equilibrium download prices. This reduces the foregone revenues from the reduction in downloads induced by advertisements, leading to higher advertising intensities. By contrast, an increase in α_j decreases equilibrium advertising intensities and increases equilibrium download prices. Finally, an increase in η_j increases download prices but does not affect advertising intensities as long as download prices are positive. Because an increase in satiation reduces usage time, advertising revenues become smaller. As a result, firms have weaker incentives to lower prices to attract consumers, leading to higher download prices. The neutrality of η_j on advertising intensities stems from the fact that usage time is proportional to η_j and that optimal advertising intensities for paid app are independent of the scale of usage times.

Tables 14 and 15 show similar comparative statics with asymmetric firms. In each case, we name the firm with a higher equilibrium download price as the "strong" firm and the firm with a lower equilibrium download price as the "weak" firm. We set the value of the equilibrium variables of the weak firm to 1 and show the relative value of other equilibrium variables with different parameters. In Table 14, apps differ in ξ_{ej}, and firms with larger values of ξ_{ej} set higher download prices and smaller advertising intensities. A larger ξ_{ej} results in a firm being more willing to collect revenues from in-app purchases rather than advertising revenues, leading to lower advertising intensities. The effect of ξ_{ej} depends on its effects on in-app purchases and market shares. Whereas an increase in ξ_{ej} increases in-app purchases, which tend to decrease download prices to attract more consumers, the increase in market share accompanying the increase in the usage value leads to the greater market power and higher download prices. In our example, the latter effect dominates and, thus, the firm with a large value of ξ_{ej} sets higher download prices. In Table 15, firms differ in the number of apps they provide, and firms with a larger set of apps set higher download prices to avoid cannibalization among apps provided by the same developer. However, firms set the same advertising intensities as those with a smaller set of app, as long as the equilibrium download prices are positive. This phenomenon results from the well-known fact in the media literature that, as long as the price can be flexibly chosen, any impact of competition on advertising intensities is neutralized by changes in download prices (Anderson and Gabszewicz, 2006).

Our model also allows monetization policies to vary depending on the ownership structure. Table 16 provides examples of equilibria under two environments in which only the ownership structures differ. One equilibrium is under a single-product duopoly, and another is under a two-product monopoly. Under a single-product duopoly, download prices are set to zero and are strictly

Table 13: Comparative statics in symmetric oligopoly

Baseline parameter, $(\alpha_y, \alpha_a, \eta) = (0.1, 0.05, 2)$:

	Baseline	$\alpha_y = 0.101$	$\alpha_a = 0.0505$	$\eta = 2.02$
Ad	1	1.0003	0.9897	1

	Baseline	$\alpha_y = 0.101$	$\alpha_a = 0.0505$	$\eta = 2.02$
price	1	0.99	1.0001	1.0001

Table 14: Comparative statics with asymmetry in ξ_e

Baseline parameter, $(\alpha_y, \alpha_a, \eta) = (0.1, 0.05, 2)$:

	Baseline	$\alpha_y = 0.101$	$\alpha_a = 0.0505$	$\eta = 2.02$
Ad (weak firm)	1.0000	1.0003	0.9897	1.0000
Ad (strong firm)	0.9953	0.9957	0.9850	0.9953

	Baseline	$\alpha_y = 0.101$	$\alpha_a = 0.0505$	$\eta = 2.02$
Price (weak firm)	1.0000	0.9900	1.0001	1.0001
Price (strong firm)	1.0001	0.9901	1.0002	1.0002

Table 15: Comparative statics with asymmetry in ownership

Baseline parameter, $(\alpha_y, \alpha_a, \eta) = (0.1, 0.05, 2)$:

compara	Baseline	$\alpha_y = 0.101$	$\alpha_a = 0.0505$	$\eta = 2.02$
Ad (weak firm)	1	1.0003	0.9897	1
Ad (strong firm)	1	1.0003	0.9897	1

	Baseline	$\alpha_y = 0.101$	$\alpha_a = 0.0505$	$\eta = 2.02$
Price (weak firm)	1.0000	0.990	1.0001	1.0001
Price (strong firm)	0.8596	0.851	0.8597	0.8597

positive under a two-product monopoly because single-product firms have greater incentives to set lower prices to attract consumers from apps provided by other firms. This example illustrates how monetization policies can vary according to changes in ownership structures, such as mergers.

Table 16: Ownership structures and monetization regimes

	Advertising intensity	Download price
Two-product monopoly	1.0484	1.3403
Single-product duopoly	1.0472	0.0000