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

Kohei Kawaguchi† Toshifumi Kuroda‡ Susumu Sato§

November 1, 2021

[Click here for the latest version]

Abstract

In the mobile app market, multiple monetizing policies such as paid and free ad-sponsored models co-exist. This paper proposes a novel model of ad-sponsored media with endogenous business model choice, that is applicable to the mobile app industry. The model defines an equilibrium over consumers' downloads and usage decisions and app developers' pricing and advertising decisions. The cost for a consumer to use an app is well defined regardless of the monetizing policy, enabling a Small, Non-transitory but Significant Increase in Cost (SSNIC) test for defining an antitrust relevant market. We estimate the model using mobile app data from Japan from 2015 to 2017. The SSNIC test shows that few non-game categories are relevant markets, whereas so are many game categories. The relevant market definition based on a full equilibrium simulation is aligned with this result. Furthermore, merger simulations show that the welfare damage is more pronounced in categories where the hypothetical monopolist's profit increases more in the SSNIC test, validating the use of SSNIC test as a convenient screening tool. Merger simulations suggest the importance of endogenous business model choice, because price increase is often caused by the shift from free to paid model. The endogenous business model choice also affects the implication of the platform transaction fee reduction. Contrary to the standard vertical relation, in the app economy, we find that a reduction in the transaction fee increases the price, because apps find it more profitable to charge price rather than show advertisements to consumers.

Keywords: Merger simulation, market definition, SSNIP, antitrust policy, ad-sponsored media, platform transaction fee, app economy, distributed word representation.

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

*This study was financially supported by JSPS grant 18K12768 and 20K22117 and NHK. At the time of the research, Kuroda was a member of NHK's Internet business committee and Japanese Fair Trade Commission (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 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, SWET, JEES, NBER SI, and Happy Hour Seminar! We thank Akira Matsushita for the research assistance. All errors are our own.

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

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

§susumusato.econ@gmail.com, 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 due to the co-existence of multiple monetizing policies such as *freemiums*. In our representative sample of top apps in Japanese Google Play, 21% of non-game apps and 27% of game apps are free ad-sponsored apps, whereas the rest charge either download price or in-app purchase to consumers. The presence of free apps prevents us from identifying substitution patterns based on a traditional method that primarily uses price variations.

The theoretical and empirical uncertainty in the definition of a relevant market and the welfare 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.[1] Although the merger was once approved, the U.S. Federal Trade Commissions (FTC) and states later requested a breakup.[2]

The policymakers are also facing a challenge with regulating the vertical relation between app developers and app marketplaces such as *App Store* and *Google Play*. *Spotify* has long complained about 30% transaction fee charged by App Store on the download revenue and in-app purchases.[3] In 2020, *Epic Games* also filed lawsuits against *Apple* and *Google* for the transaction fee.[4] However, it is theoretically unclear whether the current 30% transaction fee is too high or too low because most mobile apps can monetize either through prices or advertising. When they are charged on the non-advertising revenues, they may increase advertisements and reduce prices (Belleflamme and Toulemonde, 2018). Thus, the welfare implication depends on the market environment, such as consumers' disutility of watching advertisements.

In this paper, we propose a new framework of ad-sponsored media with endogenous business model choice and apply it to the mobile app industry. To address the co-existence of multiple monetizing policies of ad-sponsored media, 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

[1] Case No COMP/M.7217 - Facebook/ WhatsApp `https://ec.europa.eu/competition/elojade/isef/case_details.cfm?proc_code=2_M_7217`, accessed on February 12, 2021.

[2] FTC v. Facebook. `https://www.ftc.gov/enforcement/cases-proceedings/191-0134/facebook-inc-ftc-v`, accessed on February 12, 2021.

[3] Spotify's claim `https://www.timetoplayfair.com/timeline/`, accessed on February 21, 2021 and European Commision press release `https://ec.europa.eu/commission/presscorner/detail/en/IP_20_1073`, accessed on February 21, 2021.

[4] Epic Game's claim `https://www.epicgames.com/site/en-US/free-fortnite-faq`, accessed on February 21, 2021.

receives from advertisers. We explicitly model this non-price competition of mobile app developers regarding advertising intensity. An app developer's business model choice is characterized by non-negativity constraints on pricing and advertising intensity.

To facilitate analysis, following Armstrong and Vickers (2001), we transform the original model of price and advertising competition into an equivalent *competition-in-utility* model wherein app developers set the mean utilities served to consumers and then choose the optimal level of prices and advertising intensities to achieve those utilities. This transformation allows us to define the "price" or the "cost" of an app for a consumer in a unified manner, regardless of the monetizing policies, as the difference in the maximal attainable utility from the app and the achieved utility due to either download price or advertisements. This is particularly useful in the assessment of market power of ad-sponsored media facing a two-sided market.

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.

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 because of the so-called ad technology. 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.

We estimate the model using detailed data about mobile download, revenue, and usage for apps that have regularly appeared in the download or usage ranking of Google Play in Japan during 2015-2017. We selected the sample to represent the population proportion of business models. The analysis covers entire app categories including games and non-game categories. We separately estimate the model for non-game and game categories, because the shape of consumer demand and app developers' business model substantially differs between game and non-game apps. In general, downloads are small but usage time is long for game apps and downloads are large but usage time is short for non-game apps. App developers are more likely to charge price to consumer for game apps and show advertisements for non-game apps. The estimates show that advertising decreases consumers' utility. The parameter estimates indicate the disutility of watching one unit (eCPM) of advertisements is on average JPY 24-27. Because the average advertising price is JPY 425.8, these amount to 5-6% of the app's advertising revenue. These numbers are smaller than the estimate of Facebook presented in the study of Benzell and Collis (2020). Based on a survey, they estimated that the disutility of advertising is approximately 20% of Facebook's advertising revenue. The estimates also indicate that consumers are more segmented into categories for game apps than for non-game apps. As a consequence, the demand substitution is more likely closed within a product category for game apps than non-game apps.

Using the estimated model, we conduct *Small, Non-transitory but Significant Increase in Cost (SSNIC)* test one the basis of competition-in-utility model to define relevant markets in the mobile apps market, which extends the *Small, Non-transitory but Significant Increase in Price (SSNIP)*. Although it is impossible to conduct standard SSNIP test because of the prevalence of free apps, summarizing advertisements and prices as costs allows us to conduct the SSNIC test. In addition, we implement the Equilibrium Relevant Market Cost (ERMC) test, an extension of the Equilibrium Relevant Market Price (ERMP) test (Ivaldi and Lorincz, 2011). In this test, we consider a hypothetical monopolist owning a set of apps and compute the counterfactual equilibrium. The set of apps is considered to be an antitrust relevant market if the average consumer's cost increases more than a threshold, say 5%.

We test whether product categories as defined in Google Play constitute an antitrust relevant market. When using SSNIC test with a cost increase of 5%, none of non-game app categories including comics, communications, social, and so forth, does not constitute an antitrust relevant market, except for the tools category. Even for tools category, a hypothetical monopolist can barely increase the profit by 0.01% and many consumers substitute to other apps responding to the cost increase. The exogenous cost increase in the SSNIC test is especially harmful for categories like comics, communications, entertainment, and social apps, suggesting that apps in these categories compete for consumer's time beyond the category.

In contrast, some of game categories including action, casino, casual, puzzle, role playing, and simulation apps are judged as antitrust relevant markets by the SSNIC test. This stems from the low estimated consumer's substitution across categories for game apps.

The relevant market definition using the ERMC test with a threshold cost increase of 5% is largely aligned with the definition using the SSNIC test. None of the non-game categories constitute antitrust relevant markets and some game categories including action, casual, puzzle, and role playing are found to increase the average cost more than 5%. Compared with the SSNIC test, the ERMC test incorporates the strategic response of other competitors and the hypothetical monopolist's optimal pricing over owned apps. The fact that nevertheless the results of these tests are aligned would validate the use of SSNIC test for a practical purpose.

The simulation of mergers of apps in each category shows that the welfare damages to consumers are more pronounced in categories that are judged as antitrust relevant markets by the SSNIC test. If a category is not regarded as a relevant market, the welfare impact is negligible. Even in categories judged as relevant markets, the welfare damage is negligible if the hypothetical monopolist's profit increase is small. Therefore, the hypothetical monopolist's profit change in the SSNIC test is a useful sufficient statistics for screening potentially anti-competitive mergers.

The simulation of mergers among top-ranked apps shows differential effects of mergers on welfare across game and non-game apps. The merger involving top-50 non-game apps increase the profit of the app developers by 8% and reduces consumer surplus by 15.7%, whereas the merger involving top-50 game apps increase the profit of app developers by 35% and reduces consumer surplus by 6.5%. This difference in welfare effect is driven by the difference in the levels of product differentiation between game and non-game apps.

All of the merger analyses suggest that the endogenous business model choice of app developers plays an important role because large mergers shift the business model of some free apps to paid business models, which contributes to the increase in the price. This shift in the business model occurs because mergers increases the market power of marginally free apps and induces them to charge prices. This result suggests that ignoring the business model choice of app developers may fail to incorporate anti-competitive price effects of mergers involving free apps.

Finally, we conduct a counterfactual analysis by gradually reducing the platform transaction fee from 30 to 0%. In the standard vertical relation, a reduction of transaction fee reduces the price that apps charge to consumers. However, we find the opposite happens in the app economy. According to the counterfactual simulations, a reduction of transaction fee increases the median price of apps. At the same time, a reduction of transaction fee decreases the advertising intensity.

The endogenous business model choice is the key for understanding this counter-intuitive result. As the transaction fee declines, it becomes more profitable for app developers to collect revenues from download prices. Therefore, some of the ad-sponsored apps shift to paid business model. This leads to an increase in the price and a reduction in the advertisement. The endogenous business model effect dominates when a non-negligible number of free apps exists and their profitability from charging price and showing advertisement are at the margin. The estimated parameters and the distribution of the marginal cost shocks exhibit this feature, reflecting the fact in data that 21% of non-game apps and 27% of game apps has experienced pricing model change during the data period. The increase in the price is sharper for non-game apps than for game apps due to the higher share of free apps in non-game categories than in game categories.

The remainder of this paper is organized in the following manner. In the rest of this introduction, we clarify 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. Section 3 lays out the model and proposes an algorithm to solve the model. 4 derives an estimator for the key structural parameters in the model. Section 5 explains the data we use for the analysis and Section 6 describes the estimation results. Section 7 defines the relevant market for several apps based on the estimated model. Section 8 conducts hypothetical merger simulations and evaluate the competitiveness of the mobile app market. Section 9 analyzes the effect of platform fee reduction, and Section 10 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 single-home, and advertisers multi-home (Anderson and Peitz, 2020), or both consumers and advertisers 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 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) and 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 investigate the effect of product updates. Yuan (2021) estimates consumer's download and usage of a pair of mobile apps. Our paper differs from these studies in multiple dimensions. First, we consider both download and usage decisions over mobile apps. The only exceptions are Han et al. (2016) and Yuan (2021). However, Han et al. (2016)'s data and analysis are at the product category level, whereas ours is at the product level. Yuan (2021) considers a pair of apps but we cover the entire apps. 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 estimate demand, but did not consider advertising intensity. Third, we include high-dimensional product characteristics elicited from in-text product information, allowing us to avoid making an assumption about the product category to which each app belongs. 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 have studied the strategy of mobile app developers. Ghose and Han (2014) considered the price competition faced by mobile app firms. Ershov (2020) investigated 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 ranges 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 (DOJ) 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.

2 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 and provide recent competition policy related issues.

Mobile app Mobile app is an 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.[5] The market share in tablets was 29.0% for Android and 70.7% for iOS.[6]

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 browsers. 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 App Store but can be downloaded from several stores for the Android OS. Google operates the Google Play as mobile apps store, and other firms operate mobile app stores for Android devices including *Galaxy Store* for *Samsung* devices and *Epic Games* for some games such as *Fortnite*. Nevertheless, in 2017, the majority of the downloaded apps were still from Google Play and App Store. To distribute a mobile app through a mobile app stores, the developer has to pass a review process, and these processes have review policies that differ across mobile app stores.[7]

[5] Mobile operating system market share worldwide in 2017, statcounter, `http://gs.statcounter.com/os-market-share/mobile/worldwide/2017`, accessed on February 13, 2021.

[6] Tablet operating system market share worldwide in 2017, statcounter, `http://gs.statcounter.com/os-market-share/tablet/worldwide/2017`, accessed on February 13, 2021.

[7] The review guideline for App Store is available at `https://developer.apple.com/app-store/review/guidelines/` and the review guideline for Google Play is available at `https://play.google.com/intl/ja/about/developer-content-policy-print/`, accessed on February 13, 2021.

A mobile app has a page on each mobile store that provides information about the app. The information in the store is described in Section 5.2. Mobile app stores classify apps into several categories. Google play first classifies apps into *Application* and *Games*. Then, it classifies them into categories such as "Social", "Music and Audio" and "News and Magazines" for application apps, and "Action", "Puzzle" and "Sports" for game apps to enable consumers to easily find a desired app. The details of the categories are also described in Section 5.2. Figure 1 provides an example of a page on Japanese Google Play in 2018.

Mobile app developers Mobile app developers face a two-sided market of consumers and advertisers. Both sides of the market have grown 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 reported 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.

Mobile app developers also have a vertical relationship with the app stores. The app stores charge a transaction fee on the revenues from download and in-app purchases. Both Google Play and App Store charge a transaction fee of 30%, and developers earn only 70% of the download and in-app purchase revenues. Some app developers attempted to collect revenues outside the app stores. However, the review guideline of Google Play and App Store prohibit this practice.

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.[8] An ad network distributes 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,

[8] Available at `https://www.appsflyer.com/2018indexpage/`, accessed on February 13, 2021.

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.[9] 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.[10] 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, high-tech titans (Gilbert, 2020) 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. In 2021, the FTC and states in the U.S. requested a breakup and the debate is still ongoing.

Another recent merger case that involves a high-tech titan is the Google/Fitbit case. In 2019, Google announced to acquire *Fitbit* that sells health and fitness smartwatch. The European Commission and JFTC approved it with a number of conditions, such as the prohibition of the use of Fitbit health data for ad targeting.[11] In January 2021, Google announced that it had closed the deal for USD 2.1 billion, although it was under review by the U.S. DOJ and the Australia's Competition & Consumer Commission.[12]

There are also ongoing antitrust cases regarding the vertical relation between app developers and app stores. Spotify claimed that the transaction fees are used to protect *Apple Music* (Recode, 2016). In 2020, the European Commission launched a formal investigation into Apple to address

[9]See `https://support.google.com/admob/answer/3418058?hl=en`, accessed on February 13, 2021.

[10]Available at `https://japan.inmobi.com/advertising-cloud/mediation`, accessed on February 13, 2021.

[11]European Commission press release is available at `https://ec.europa.eu/commission/presscorner/detail/en/ip_20_2484`, and JFTC press release is available at `https://www.jftc.go.jp/en/pressreleases/yearly-2021/January/210114.html`, accessed on February 13, 2021.

[12]See `https://www.theverge.com/2021/1/14/22188428/google-fitbit-acquisition-completed-approved`

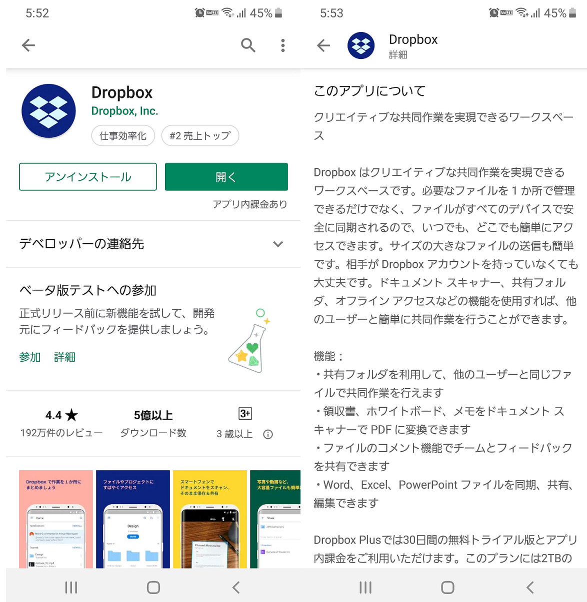


Figure 1: Product description in Google Play

Spotify's claim.[13] In 2020, Epic Games also took a legal action on Apple and Google's restrictions on app stores in the U.S., Australia and the European Union.[14]

Apple and Google reduced transaction fees to 15% for consumers whose subscription terms went beyond 1 year in 2016 and 2018, respectively.[15][16] Furthermore, Apple introduced "App Store Small Business Program," which reduces transaction fee to 15% for small businesses earning up to $1 million per year in 2021.[17]

3 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.

[13] European Commission press release is available at `https://ec.europa.eu/commission/presscorner/detail/en/IP_20_1073`, accessed on February 21, 2021.

[14] Epic Game's claim `https://www.epicgames.com/site/en-US/free-fortnite-faq`, accessed on February 21, 2021.

[15] Available at `https://support.google.com/googleplay/android-developer/answer/112622?hl=en`, accessed on February 13, 2021.

[16] The news report Apple's reduction of transaction fee. See `https://techcrunch.com/2016/06/08/apple-to-introduce-search-ads-on-app-store-along-with-changes-to-app-review-discovery-and-splits/`, accessed on February 13, 2021.

[17] See `https://www.apple.com/newsroom/2020/11/apple-announces-app-store-small-business-program/`, accessed on February 13, 2021.

3.1 Setting

Population and covariates Consider a market with a set of apps $\mathcal{J} := \{1, \ldots, J\}$ provided by a group of app developers $\mathcal{D} := \{1, \ldots, D\}$. 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, $q_j \in \mathbb{R}_+$. 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:

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

where

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

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

$$v_j(q_j, a_j, X_{uj}, \xi_{uj}) := \kappa \left\{[\beta_u' X_{uj} - \alpha_a a_j - \alpha_y w + \xi_{uj}] q_j - \psi_j(q_j)\right\}, \tag{3}$$

where ε_{ij} is an idiosyncratic taste shock distributed according to an i.i.d. type-I extreme-value distribution. The parameter κ determines the importance of usage-related utility when deciding download. We interpret this as the expected number of weeks a consumer uses an app. Equivalently, we can make the coefficients of w flexible, which is currently restricted to be α_y, but we adopt this parameterization for the sake of interpretation.

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_a is the disutility from being revealed to a unit advertisement in app j. 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 ψ as follows:

$$\psi_j(q_j) \coloneqq \frac{\eta}{2} q_j^2, \tag{5}$$

where $\eta \in \mathbb{R}_+$ is the degree of satiation from usage. Because the model becomes numerically unstable as η approaches 0, we put a lower bound on it by adding 0.01. The estimate shows that this lower-bound is not binding.

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

$$S_j = \max_{q_j} \kappa \left[\left(\beta_u' X_{uj} - \alpha_a a_j - \alpha_y w + \xi_{uj} \right) q_j - \frac{\eta}{2} q_j^2 \right], \tag{6}$$

and the indirect utility takes the following form:

$$\begin{aligned} u_{ij} &= S_j - \alpha_y F_j + \beta_{di}' X_{dj} + \xi_{dj} + \varepsilon_{ij} \\ &\coloneqq \delta_j + \nu_i' \Sigma X_{dj} + \varepsilon_{ij}, \end{aligned} \tag{7}$$

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

3.2 Consumer's Problem

Usage decision Next, we solve the consumer problem. Let $X_j = (X_{uj}', X_{dj}')'$, $\xi_j = (\xi_{uj}, \xi_{dj})$, $\theta_i = (\alpha_a, \beta_u', \alpha_y, \eta, \beta_{di}')'$, and $\theta = (\alpha_a, \beta_u', \alpha_y, \eta, \beta_d', \mathrm{vec}(\Sigma)')'$. By solving the first-order condition for usage q_j, we obtain:

$$q_j = \tilde{q}_j(a_j, w, X_j, \xi_j; \theta) \coloneqq \max \left\{ \frac{1}{\eta} \left(\beta_u' X_{uj} - \alpha_a a_j - \alpha_y w + \xi_{uj} \right), 0 \right\}. \tag{8}$$

By substituting equation (8) into equation (6), we obtain the following usage surplus function:

$$S_j = S(a_j, w, X_j, \xi_j; \theta) \coloneqq \frac{\kappa\eta}{2} \tilde{q}_j^2(a_j, w, X_j, \xi_j; \theta), \tag{9}$$

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

$$\begin{aligned}\delta_j =&\delta_j(a_j, F_j, w, X_j, \xi_j; \theta)\\ :=&S(a_j, w, X_j, \xi_j; \theta) + \beta_d' X_{dj} - \alpha_y F_j + \xi_{dj}.\end{aligned} \tag{10}$$

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:

$$s_j = \tilde{s}_j(a, F, w, X, \xi; \theta) := \int_{\mathbb{R}^{K_d}} \frac{\exp\left[\delta_j(a_j, F_j, w, X_j, \xi_j; \theta) + \nu_i' \Sigma X_{dj}\right]}{1 + \sum_k \exp\left[\delta_k(a_k, F_k, w, X_k, \xi_k; \theta) + \nu_i' \Sigma X_{dk}\right]} dG_{\nu_i}(\nu_i). \tag{11}$$

3.3 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:

$$\Pi_d(a, F, X, w, \xi; \theta) := \sum_{j\in\mathcal{J}_d} \pi_j(a, F, X, w, \xi; \theta), \tag{12}$$

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

$$\begin{aligned}&\pi_j(a, F, X, w, \xi; \theta)\\ :=&s_j(a, F, X, w, \xi; \theta) \left\{(1-\rho)F_j + q_j(a, F, X, w, \xi; \theta)(a_j r - \lambda) - \epsilon_j\right\},\end{aligned} \tag{13}$$

where r is the advertising revenue per unit of advertisements shown to the consumer, ρ is the transaction fee 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, λ is marginal cost for the usage, and ϵ_j is other marginal cost for the download. We assume that ϵ_j is an i.i.d. random variable drawn from distribution G_ϵ.

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 to show advertisements as one of the product characteristics of a 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:

$$\max_{\{(a_j,F_j)\}_{j\in\mathcal{J}_d}} \Pi_d(a,F,X,\xi;\theta) \tag{14}$$

$$\text{s. t.} \quad a_j \geq 0, \quad j \in \mathcal{J}_d, \tag{15}$$

$$F_j \geq 0, \quad j \in \mathcal{J}_d. \tag{16}$$

The first-order conditions for this problem are:

$$\frac{\partial \Pi_d}{\partial F_j} := (1-\rho)s_j + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial F_j}[(1-\rho)F_k + q_k(a_k r - \lambda) - \epsilon_k] \leq 0, \tag{17}$$

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

$$\begin{aligned}\frac{\partial \Pi_d}{\partial a_j} := {} & s_j q_j r + s_j \frac{\partial q_j}{\partial a_j}(a_j r - \lambda) \\ & + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial a_j}[(1-\rho)F_k + q_k(a_k r - \lambda) - \epsilon_k] \\ & \leq 0,\end{aligned} \tag{18}$$

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

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

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 (17) and (18).

3.4 Conversion to an equivalent competition-in-utility model

We further transform the original model of competition in advertising intensities and prices into an equivalent *competition-in-utility* model (Armstrong and Vickers, 2001). By doing so, we can define "cost" for a consumer to use an app, which includes the cost of paying money and digesting advertisements. Then, we can conduct a hypothetical monopolist test based on the "cost" when the Small but Significant Non-transitory Increase in Price (SSNIP) test is infeasible because of zero price products. As a side product, we can speed up the computation of an equilibrium. We show the overview of the analysis below, and the detail can be found in Appendix A.

Suppose that each developer is constrained to achieve the mean utility δ_j for app j. Then, the developer chooses the optimal pair (a_j, F_j) of advertising intensity and download price to achieve δ_j. With this pair, the developer earns per-consumer profit $\bar{\pi}_j(\delta_j)$ by offering mean utility δ_j. Note that $\delta_j \in (-\infty, \delta_j^0]$, where δ_j^0 is the maximum mean utility that app j can provide, which is achieved when app j has zero download prices and advertising intensities. Then, given a profile of mean utilities $\delta = (\delta_j)_{j \in \mathcal{J}}$, and the profit earned by app j is given by $s_j(\delta) \times \pi_j(\delta_j)$. Therefore, each developer d's profit can be written in the following manner:

$$\bar{\Pi}_d = \sum_{j \in \mathcal{J}_d} s_j(\delta)\bar{\pi}_j(\delta_j),$$

which has the partial derivative

$$\frac{\partial \bar{\Pi}_d}{\partial \delta_k} = s_k(\delta)\bar{\pi}'_k(\delta_k) + \sum_{j \in \mathcal{J}_d} \frac{\partial s_j(\delta)}{\partial \delta_k}\bar{\pi}_j(\delta_j).$$

Thus, solving the first-order condition

$$\frac{\partial \bar{\Pi}_d}{\partial \delta_k} \leq 0$$

with equality if $\delta_j < \delta_j^0$ for $j \in \mathcal{J}_d$ and $d \in \mathcal{D}$ gives the equilibrium mean utility δ_j^* for $j \in \mathcal{J}$. Finally, computing the optimal pair (a_j, F_j) of advertising intensity and download price of app j that achieves mean utility δ_j, we can separate the equilibrium advertising intensities and download prices.

Under the competition-in-utility framework, we can define the notion of cost c_j of app j as the gap between its maximal mean utility δ_j^0 and its actual mean utility δ_j, that is, $c_j = \delta_j^0 - \delta_j$. For example, when an app j is ad-free and only charges download price F_j, the cost is given by $c_j = \alpha_y F_j$. Setting the cost c_j leads to the mean utility $\delta_j = \delta_j^0 - c_j$ and gives the per-consumer profit $\bar{\pi}_j(\delta_j^0 - c_j)$ in our competition-in-utility framework. We use the notion of cost for conducting a hypothetical monopolist test in Section 7.

3.5 Discussion

Non-price competition This framework allows for analyzing non-price competition by treating mobile apps as ad-sponsored media that compete in prices and the amounts of advertisements shown to consumers. This feature make it possible to conduct an anti-trust analysis of free apps that compete in variables other than prices. For example, when we consider a merger between free apps, prices may not increase in the framework of price competition. However, within our framework, either app's price or advertising intensity increases following a merger. We illustrate competitive effects of mergers in Section 8.

Particularly, the effects of competitive pressures on prices, advertising intensities, and usage levels depends on the business model of each app. When an app is paid and ad-sponsored, competitive pressure affects prices but not advertising intensities. This is observed by combining the optimality conditions (17) and (18), which derives the condition for the optimal advertising intensity:

$$\frac{\partial q_j}{\partial a_j}\left[\frac{(1-\rho)}{\alpha_y}\kappa\eta q_j + a_j r - \lambda\right] + q_j r = 0.$$

This equation is determined solely by the characteristic of app j, thereby independent of competitive pressure. By contrast, when an app is free but ad-sponsored, competitive pressures affects advertising intensities instead of prices, because advertising intensities are determined by single optimality condition (18).

In the merger analysis, these differential effects of competitive pressures on prices and advertising intensities in the following manner; a merger raises prices of paid apps and advertising intensities of free apps. Furthermore, free apps may shift to paid apps following a merger, in which case the price rather than advertising intensities would increase. Thus, empirically, the degree of slackness of the first-order condition (17) for free apps determines the effects of mergers on prices and advertising intensities. This is captured by a parameter χ introduced in Section 4.

Endogenous business model The model features endogenous business models. In particular, business models are determined by comparing the profitability of hosting advertisements relative to pricing, which depends on underlying parameters, cost structures, and transaction fees set by the platform. In Section 9, we show that this endogenous business model choice is essential to understand the effects of transaction fees.

An increase in the transaction fee has two effects on prices. First, the lower the transaction fee is, the more it is profitable for app developers to collect revenues from download prices. This leads to an increase prices and reduction in advertisements because of substitution. Second, because the transaction fee takes the form of a proportional fee, when the "effective marginal cost," the marginal cost minus advertising revenue, is negative, the presence of proportional fee can reduce the prices. To illustrate, suppose that a single-product price-setting firm with negative effective marginal cost $-\psi < 0$ faces a demand function $d(p)$ and proportional fee ρ, its profit is given by

$$d(p)[(1-\rho)p+\psi] = (1-\rho)d(p)\left[p+\frac{\psi}{1-\rho}\right].$$

Then, the app developer optimally sets the price that satisfies

$$p = \left|\frac{d(p)}{d'(p)}\right| - \frac{\psi}{1-\rho}.$$

We can see that p decreases with ρ as long as $\psi > 0$, whereas p increases with ρ when $\psi < 0$, which is the case when marginal cost is positive.

Put these together, an increase in transaction fees can increase or decrease with transaction fees depending on the sign of effective marginal costs. In particular, the distribution of marginal cost G_ϵ, the advertising revenue r, and usage-related parameters. For example, when marginal cost are low and advertising revenue is high, an increase in transaction fee lowers price.

Limitations It will be worth highlighting a few limitations of the model. First, the model assumes that mobile apps face inelastic advertiser demand at exogenously and commonly given advertising price r. This presumes that mobile apps have no market power against advertisers and the value of advertising slots is homogeneous across apps. In reality, apps would have some market powers and differ in the attractiveness for advertisers. Ignoring these features understate some of the key feature in the two-sided market environment, such as "sea-saw effects" that says a change in the market environment hurting consumers can benefit advertisers (Anderson and Peitz, 2020).

Second, the model assumes that download, subscription, and in-app purchase revenues and advertisement revenues account for the entire revenue of an app developer. In reality, app developers may obtain additional revenues by directly transacting with consumers outside the app store and by selling consumer data to third-party entities. In the estimation, these additional revenues, if exist, will be attributed to the negative marginal cost shocks.

Finally, the model assumes away consumer heterogeneity in usage-related utility. This assumption allows us to well define the costs of apps to consumers. Without this assumption, the consumer's cost of an app can be different across consumers and the SSNIC test is not straightforward. This assumption, on the other hand, prohibits us from analyzing an app developer's versioning strategy as price discrimination. Given that a non-negligible number of apps have basic and premium versions, ignoring consumer heterogeneity in usage utility can affect the distributional effects of competitive pressures on consumers of low-end apps and high-end apps.

4 Estimation

4.1 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_{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 $\theta := (\theta_d', \theta_u', \lambda', \iota')'$, where $\theta_d := (\alpha_y, \eta, \beta_d', \text{vec}(\Sigma)')'$ 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 $\theta_0 := (\theta_{d0}', \theta_{u0}', \lambda_0', \iota_0')'$ denote true parameters.

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

$$s_j = \int_{\mathbb{R}^{K_d}} \frac{\exp\left[\delta_j(a_j, F_j, w, X_j, \xi_j; \theta) + \nu_i' \Sigma X_{dj}\right]}{1 + \sum_k \exp\left[\delta_k(a_k, F_k, w, X_k, \xi_k; \theta) + \nu_i' \Sigma X_{dk}\right]} dG_{\nu_i}(\nu_i). \tag{19}$$

Although this equation involves 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:

$$\begin{aligned} \delta_j(a_j, F_j, w, X_j, \xi_j; \theta) &= S(a_j, w, X_j, \xi_j; \theta) + \beta_d' X_{dj} - \alpha_y F_j + \xi_{dj} \\ &= \frac{\kappa\eta}{2} q_j^2 + \beta_d' X_{dj} - \alpha_y F_j + \xi_{dj}. \end{aligned} \tag{20}$$

By inserting equation (20) into equation (19), 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 in equation (20). This dependence results from the functional-form assumption in (8), 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 (20) 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 (19) 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 and ϵ Next, we elicit advertising intensity $\{a_j\}_{j\in\mathcal{J}}$ and download-related marginal costs ϵ. To elicit the advertising intensity $\{a_j\}_{j\in\mathcal{J}}$, we utilize the first-order conditions for advertising intensity:

$$s_j q_j r + s_j \frac{\partial q_j}{\partial a_j}(a_j r - \lambda) + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial a_j}\{(1-\rho)F_k + q_k(a_k r - \lambda) - \epsilon_k\} \leq 0, \tag{21}$$

where

$$\frac{\partial q_j}{\partial a_j} = -\frac{\alpha_a}{\eta}, \tag{22}$$

and

$$\frac{\partial s_k}{\partial a_j} = \begin{cases} -\kappa\alpha_a q_j \int_{\mathbb{R}^{K_d}} \frac{\exp(\delta_k+\nu_i'\Sigma X_{dk'})}{\left[1+\sum_{k'}\exp(\delta_k+\nu_i'\Sigma X_{dk'})\right]}\left[1-\frac{\exp(\delta_k+\nu_i'\Sigma X_{dk'})}{\left[1+\sum_{k'}\exp(\delta_k+\nu_i'\Sigma X_{dk'})\right]}\right] dG_{\nu_i}(\nu_i) & \text{for } k=j \\ -\kappa\alpha_a q_j \int_{\mathbb{R}^{K_d}} \frac{\exp(\delta_j+\nu_i'\Sigma X_{dj})\exp(\delta_k+\nu_i'\Sigma X_{dk'})}{\left[1+\sum_{k'}\exp(\delta_k+\nu_i'\Sigma X_{dk'})\right]^2} dG_{\nu_i}(\nu_i) & \text{for } k\neq j \end{cases} \tag{23}$$

for all $j \in \mathcal{J}$. Given the data $(s_j, q_j, p_j)_{j\in\mathcal{J}}$ and computed values of $(\xi_{dj}(\theta))_{j\in\mathcal{J}}$, we can compute the simulated value of $\partial s_k/\partial a_j$.

We also elicit the download-related marginal costs using the pricing first-order-condition implied by the estimated parameters. To simultaneously elicit the download-related marginal costs $(\epsilon_j)_{j\in\mathcal{J}}$ and advertising intensities $(a_j)_{j\in\mathcal{J}}$, we use the equilibrium conditions for both prices and advertisements.

First, we use the equilibrium condition for prices (17) to elicit the "effective marginal cost" $\widetilde{mc}_j := \epsilon_j - q_j a_j r$. To do this we rewrite the equilibrium condition (17) as

$$(1-\rho)s_j + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial F_j}\left[(1-\rho)F_k - q_k\lambda - \widetilde{mc}_k\right] \begin{cases} = 0 & \text{if } F_j > 0 \\ < 0 & \text{if } F_j = 0. \end{cases}$$

Because the first-order conditions for free apps are given in the form of inequalities, we cannot derive the exact values of $\widetilde{mc}_j$ for free apps by using the system of first-order conditions alone.

To elicit the exact value of marginal costs that is consistent with the equilibrium conditions, we adopt the following assumption. For apps with positive download prices, we use the equality first-order condition (17). For apps with zero download prices, we assume that if the marginal costs of free apps were greater by χ, the first-order condition (17) would be satisfied with equality. In other words, χ is the average slackness in the pricing first-order condition for free apps.

Then, the elicited value of the effective marginal cost $\widetilde{mc}_j$ satisfies

$$(1-\rho)s_j + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial F_j}\left[(1-\rho)F_k - q_k\lambda - \widetilde{mc}_j - 1\{F_k = 0\}\chi\right] = 0 \tag{24}$$

for $j \in \mathcal{J}$. Solving this system of equations yields the elicited values of effective marginal costs $\widetilde{mc}_j$, $j \in \mathcal{J}$.

Using the elicited effective marginal costs, we find the advertising intensities that are consistent with the equilibrium condition of advertisements (21). Using equation (24), equation (21) can be rewritten as below:

$$s_j \left[q_j r + \frac{\partial q_j}{\partial a_j}(a_j r - \lambda) - (1-\rho)\frac{\kappa q_j \alpha_a}{\alpha_y} \right] + \frac{\kappa q_j \alpha_a}{\alpha_y} \sum_{k \in \mathcal{J}_d} \frac{\partial s_k}{\partial F_j} 1\{F_k = 0\}\chi \leq 0, \tag{25}$$

which holds as equality if $a_j > 0$. We can solve this equation for each a_j because these equations are independent of each other for given parameter values and observed data (s, q, e, X_d). Finally, using the equation $\widetilde{mc}_j = \epsilon_j - q_j a_j r$, we find the value of download-related marginal cost ϵ_j.

We identify χ based on the assumption that the marginal costs of free and paid versions of an app are the same. Therefore, estimated χ is the average slackness in the first-order condition of the free app of a pair of free and paid versions of the same app.

Solving for ξ_u By plugging $\xi_e(\theta_d)$ and $a(\theta)$ into the first-order condition for usage (8), we obtained the implied value of ξ_{uj}:

$$\xi_{uj} = \eta q_j + \alpha_a a_j(\theta) + \alpha_y w - \beta_u' X_{uj}. \tag{26}$$

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:

$$\mathbb{E}[\xi_{uj}(\theta_0)|Z_{uj}] = \mathbb{E}\left[\xi_{dj}(\theta_{d0})|Z_{dj}\right] = 0, \tag{27}$$

which implies:

$$\mathbb{E}[\xi_{uj}(\theta_0)Z_{uj}] = \mathbb{E}\left[\xi_{dj}(\theta_{d0})Z_{dj}\right] = 0, \tag{28}$$

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 $(X_{ujt}, X_{djt}, Z_{ujt}, Z_{djt}, F_{jt}, q_{jt}, s_{tj})$ the list of variables regarding app j in market t. Let $N := \sum_{t=1}^{T} J_t$. Let:

$$\xi_{ut}(\theta) := [\xi_{u1t}(\theta), \cdots, \xi_{uJ_t t}(\theta)]', \xi_{dt}(\theta_d) := [\xi_{d1t}(\theta_d), \cdots, \xi_{dJ_t t}]', \tag{29}$$

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

$$\xi_u(\theta) := [\xi_{u1}(\theta)', \cdots, \xi_{uT}(\theta)']', \xi_d(\theta_d) := [\xi_{d1}(\theta_d)', \cdots, \xi_{dT}(\theta_d)']', \tag{30}$$

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

Similarly, for instrumental variables and product characteristics, let:

$$Z_{\iota t} := \begin{pmatrix} Z'_{\iota 1t} \\ \vdots \\ Z'_{\iota J_t t} \end{pmatrix}, X_{\iota t} := \begin{pmatrix} X'_{\iota 1t} \\ \vdots \\ X'_{\iota J_t t} \end{pmatrix}, \iota \in \{u, d\}, \tag{31}$$

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:

$$Z_\iota := \begin{pmatrix} Z_{\iota 1} \\ \vdots \\ Z_{\iota T} \end{pmatrix}, X_\iota := \begin{pmatrix} X_{\iota 1} \\ \vdots \\ X_{\iota T} \end{pmatrix}, \iota \in \{u, d\}, \tag{32}$$

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:

$$g^D(\theta) := \frac{1}{\sum_{t \in \mathcal{T}} J_t} \begin{pmatrix} Z'_u \xi_u(\theta) \\ Z'_d \xi_d(\theta_d) \end{pmatrix} \tag{33}$$

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

4.2 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:

$$\epsilon^P_{jt}(\theta) := \frac{\partial \Pi_{jt}(\theta)}{\partial F_{jt}} 1\{F_{jt} > 0\} + \max\left\{\frac{\partial \Pi_{jt}(\theta)}{\partial F_{jt}}, 0\right\} 1\{F_{jt} = 0\} = 0. \tag{34}$$

We construct a corresponding moment such as:

$$g^P(\theta) := \frac{1}{\sum_{t \in \mathcal{T}} J_t} Z'_u \epsilon^P(\theta), \tag{35}$$

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

Cost conditions for freemium apps Some developer offers the free and paid versions of the same apps that differ in qualities and prices. To identify slackness parameter χ, we impose that the marginal costs of free and paid versions are the same . In particular, let $\mathcal{F}_t$ be the partition of $\mathcal{J}_t$, where each $f \in \mathcal{F}_t$ contains the different versions of the same app. Then, for all $j \in f \in \mathcal{F}_t$, we should have

$$\epsilon^F_{jt} := \epsilon_j(\theta) - \frac{\sum_{k \in f} \epsilon_k(\theta)}{|f|} = 0.$$

We construct a corresponding moment such as:

$$g^F(\theta) \coloneqq \frac{1}{\sum_{t \in \mathcal{T}} J_t} Z_u' \epsilon^F(\theta) \tag{36}$$

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:

$$\epsilon_{jt}^A(\theta) \coloneqq [A_{jt} - 1\{a_{jt}(\theta) > 0\}] = 0, \tag{37}$$

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:

$$g^A(\theta) \coloneqq \frac{1}{\sum_{t \in \mathcal{T}} J_t} Z_u' \epsilon^A(\theta), \tag{38}$$

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

4.3 Generalized Method-of-Moments Estimator

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

$$\hat{\theta} \ \in \mathrm{argmin}_{\theta \in \Theta} g(\theta)' \Phi^{-1} g(\theta), \tag{39}$$

where $g(\theta) \coloneqq [g^{D\prime}(\theta), g^{P\prime}(\theta), g^{A\prime}(\theta)]'$ 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 α_a, η, λ. 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.

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:

$$d_{\iota jkt} \coloneqq \sqrt{\sum_{l=1}^{K_\iota} (X_{\iota jlt} - X_{\iota klt})^2}, \tag{40}$$

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

$$\frac{1}{J_{d(j)t}} \sum_{k \in \mathcal{J}_{d(j)t}} d_{\iota jkt}, \frac{1}{J_t - J_{d(j)t}} \sum_{k \in \mathcal{J}_t \setminus \mathcal{J}_{d(j)t}} d_{\iota jkt}, \frac{1}{J_t - 1} \sum_{k \in \mathcal{J}_t} \left(d_{\iota jkt} - \frac{1}{J_t} \sum_{k \in \mathcal{J}_t} d_{\iota jkt} \right)^2 . \tag{41}$$

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.

In our framework, differentiation instrumental variables work in the following manner. As discussed in Section 3.5, competitive pressures affect both download prices and advertising intensities. Then, both download prices and advertising intensities affect prices, and the latter also affect usage. Through this reasoning, the differential instrumental variables work as instrumental variables for download and usage.

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 (38), given the rest of the parameters. Specifically, the linear parameters in θ_d and θ_u in our framework are $\theta_{d1} \coloneqq \beta_d$ and $\beta_{u1} \coloneqq \beta_u$, and the non-linear parameters in θ_d and θ_u in our framework are $\theta_{d2} \coloneqq [\alpha_y, \eta, \mathrm{vec}(\Sigma)']'$ and $\theta_{u2} \coloneqq \alpha_a$. Given $\theta_2 \coloneqq (\theta_{2d}', \theta_{2u}', \lambda)'$, the residuals in the demand-related moment condition are written as:

$$\begin{aligned} \xi_d(\theta) &= y_d(\theta) - X_d \beta_d, \\ \xi_u(\theta) &= y_u(\theta) - X_u \beta_u, \end{aligned} \tag{42}$$

with:

$$\begin{aligned} y_d(\theta) &\coloneqq \delta(\theta) - \frac{\kappa \eta}{2} q^2 + \alpha_y F, \\ y_u(\theta) &\coloneqq \eta q + \alpha_a a(\theta) + \alpha_y w, \end{aligned} \tag{43}$$

where $\delta(\theta), q, e, F, a(\theta)$, and w 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 $\delta(\theta)$ and $a(\theta)$. However, in our specification of the model, $\delta(\theta)$ and $a(\theta)$ are independent of θ_1 conditional on the observables $(s, q, X_d) = \{(s_j, q_j, X_{dj})_{j \in \mathcal{J}_t}\}_{t=1,\ldots,T}$ and non-linear parameters θ_2 for the following reasons. First, equation (19) implies that $\delta(\theta)$ can be computed only using (s, X_d), denoted by $\hat{\delta}(\theta_2, s, X_d)$. Similarly, equations (21), (22), and (23) jointly imply that $a(\theta)$ can be computed only using variables (s, q, X_d), $\delta(\theta)$, and non-linear parameters θ_2, denoted by $\hat{a}(\theta_2, s, q, X_d)$. Therefore, $y_d(\theta)$ can be evaluated as $\hat{y}_d(\theta_2, s, q, F, X_d)$ and $y_u(\theta)$ can be evaluated as $\hat{y}_u(\theta_2, s, q, w, X_d)$. Similarly, $g^P(\theta)$ can be evaluated by (s, q, F, r), λ, and $\hat{a}(\theta_2, s, q, X_d)$, denoted by $\widehat{g}^P(\theta_2, s, q, F, r, X_d)$. Finally, $g^A(\theta)$ is evaluated only with $\hat{a}(\theta_2, s, q, X_d)$, denoted by $\widehat{g}^A(\theta_2, s, q, X_d)$.

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:

$$\hat{\theta}_1(\theta_2) = \left(X' Z \Phi^{D-1} Z' X\right)^{-1} X' Z \Phi^{D-1} Z' \hat{y}(\theta_2, s, q, F, r, w, X_d), \tag{44}$$

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

4.4 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:

$$\begin{aligned} \xi_{dj} &= \gamma_d' W_j + \Delta\xi_{dj}, \\ \xi_{uj} &= \gamma_u' W_j + \Delta\xi_{uj}, \end{aligned} \tag{45}$$

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.

5 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 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.

5.1 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.

Formally, we group the apps by business models $\mathcal{G} = \{\text{free/ad}, \text{paid/no-ad}, \text{paid/ad}\}$ with shares $\{s_g\}_{g \in \mathcal{G}}$ in the original data. For each weak t, group g, and app i, we can calculate the usage/download ranking $r_i^k(g)$, where $k\{\text{usage}, \text{download}\}$. Then, we count the following number

$$n_i = \sum_{k \in \{\text{usage,download}\}} \sum_t 1 \left\{ r_i^k(g) \leq R(g) \right\},$$

where

$$R(g) = \begin{cases} C_R & \text{if } g = \text{ free/ad}, \\ C_R \times \frac{s_g}{s_{\text{free/ad}}} & \text{otherwise}, \end{cases}$$

and select the app i if $n_i \geq C_N$. We set $C_R = 100$ and $C_N = 10$ in the selection procedure. Note that if the distributions of the ranking and business models are the same across weeks, this selection procedure preserves the distribution of business models. In this case, we can regard the selected sample as a representative sample of the original data.

For each app selected using 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.

To reflect the fact that an app's revenues include the revenues from both downloads and in-app purchases, we treat the per-download gross revenue as the per-download price of an app, which is computed by dividing gross revenue 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. 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 and evolves steadily, we conduct the following calculation to modify the market sizes. We first multiply the estimated number of active devices in all periods by a constant number. We then calculate the total download

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

	N	Mean	SD	Median	Min	Max
Application						
Usage time (Hour)	28164	1.3	1.1	0.9	0.5	13.7
Download	28164	12984.2	21775.1	6032.5	1.0	369601.0
Download price (JPY)	28164	123.1	267.7	0.0	0.0	886.4
Game						
Usage time (Hour/User)	21203	3.8	3.0	3.3	0.5	26.7
Download	21203	9427.0	17459.1	4256.0	2.0	537098.0
Download price (JPY)	21203	2492.8	3745.6	704.1	0.0	12404.0

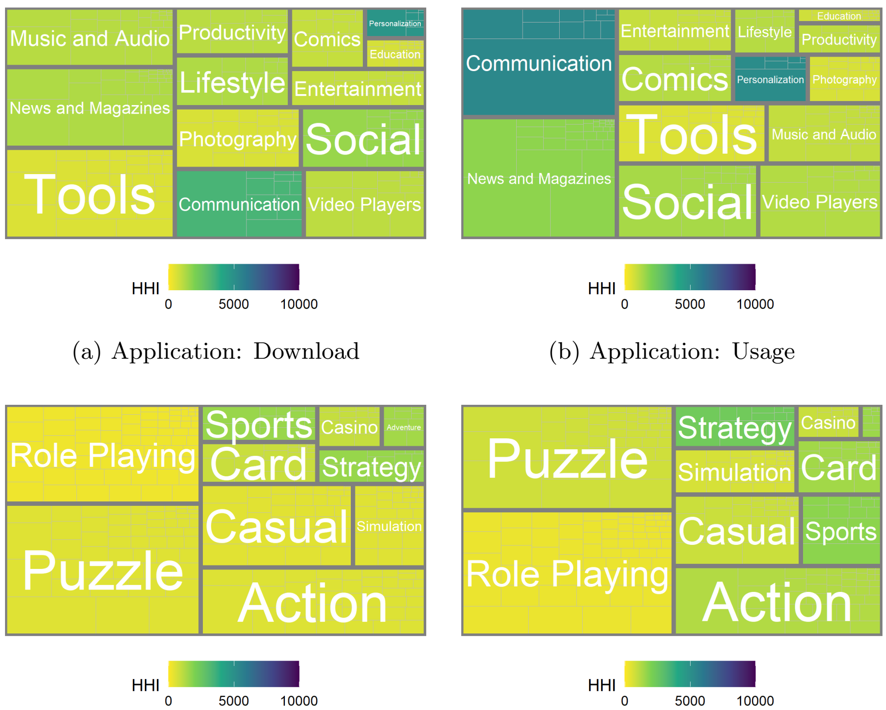
share in each market as the sum of downloads of all apps in the sample divided by the number of active devices. Based on the calculation of total download shares, we compute the trend of the total download shares. Finally, we replace the market size of each market with the one that fits the trend in the total download shares.

Summary statistics Table 1 provides the summary statistics for usage time, in-app purchase per download, number of downloads, and download price. The average usage time and in-app purchases are higher among games (3.8 h and JPY 2,492 per week) than among applications (1.3 h and JPY 123 per week). However, number of downloads is higher among applications (12,984 per week) than games (9,427 per week).

Figure 2 demonstrates the market share and Herfindahl-Hirschman Index (HHI) based on the Google Store categories. The upper panel is for applications and the lower panel is for games. Figure 3 summarizes the time series for advertising price, hourly wage, and market size. It indicates that market share based on downloads and usage can be substantially different. In terms of the download, the market structure of each category is between competitive and mildly concentrated.

5.2 Product Description

Source We use product descriptions displayed on Google Play to construct the advertising dummy, product class, and semantic vectors. We use App Annie's record of app descriptions and html files of apps in Google Play ,which was obtained by scraping the websites of Google Play as of December 2018. App Annie records a history of product descriptions in one language from among English, Japanese and other languages. It records descriptions of apps that are deleted from Google Play. The html files scraped from Google Play contains descriptions of apps written by app developers in several languages. It also contains an automated translation made by Google. To obtain as much information as possible from the description of apps, when it is available, we use Japanese descriptions written by app developers in html files. When it is not available, we use machine translation of descriptions written by app developers in other languages. For apps that have been removed from Google Play, we use App Annie's latest records. For the apps that App



(a) Application: Download

(b) Application: Usage

(c) Game: Download

(d) Game: Usage

Figure 2: Market share and Herfindahl–Hirschman Index (HHI) defined by Google Store category

Table 2: Share of app-week by business models across categories

(a) Application

Category	N	Paid/Ad sponsored	Paid/Ad free	Free/Ad sponsored
Comics	1171	0.693	0.081	0.225
Communication	1296	0.255	0.275	0.470
Education	1988	0.082	0.508	0.409
Entertainment	1375	0.255	0.131	0.615
Lifestyle	1113	0.092	0.081	0.827
Music and Audio	3238	0.148	0.311	0.540
News and Magazines	4191	0.026	0.072	0.902
Personalization	646	0.173	0.115	0.712
Photography	1853	0.131	0.107	0.761
Productivity	1204	0.098	0.425	0.477
Social	1649	0.534	0.136	0.329
Tools	2241	0.124	0.007	0.869
Video Players	1612	0.093	0.223	0.684
Total	23577	0.175	0.188	0.637

(b) Game

Category	N	Paid/Ad sponsored	Paid/Ad free	Free/Ad sponsored
Action	2757	0.710	0.032	0.258
Adventure	433	0.277	0.000	0.723
Card	1428	0.475	0.107	0.417
Casino	1216	0.604	0.002	0.395
Casual	2108	0.581	0.046	0.372
Puzzle	3829	0.630	0.000	0.370
Role Playing	3707	0.792	0.197	0.011
Simulation	1944	0.743	0.122	0.135
Sports	988	0.809	0.091	0.100
Strategy	571	0.680	0.320	0.000
Total	18981	0.669	0.083	0.248

Annie recorded in Japanese, we use them. For the apps that App Annie recorded in other languages than Japanese, we use the machine translation of them. We employ *Microsoft Azure Translator* to machine translate app descriptions that are written in other languages.

App Category Google Play classifies apps into 49 categories: 32 are application apps and 17 are game apps. However, as a result of the selection of apps, only 10 categories in application apps and 9 in game apps are used. The other categories are aggregated into "Others" and used as the base category. We use those categories as the product category dummy.

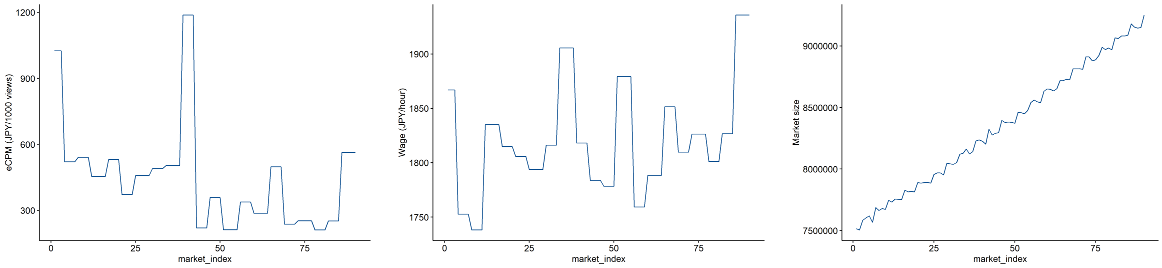
However, as app categories in Google Play are set by app developers[18], the rules for categorizing apps are not uniform. Similar apps can be classified into different categories and apps with completely different features can be put in a category. Therefore, metrics that quantify app characteristics in more detail than categories are required to capture the substitution pattern of apps.

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.

Business model Table 2 represents the share of apps showing advertisements and download prices charged in each category at the app and week level. Because the advertisement dummies are time invariant, the time dimension is only relevant for pricing. These are the proportion of apps selected for estimation. The apps are selected so as to approximately replicate the population business model proportion.

The table underscores the co-existence of business model in the app economy. There are several interesting findings in this table. First, there are non-negligible share of paid apps even among non-game categories once we include in-app purchases into the price. 36.3% of non-games apps and 75.2% of game apps charge price on consumers. The paid model is not only for games and the co-existence of business model is prevailing in both non-game and game apps. Second, the proportion of free/ad sponsored apps of non-game apps is 63.7% and it is higher than a proportion of 24.8% of game apps. Therefore, non-games apps are more likely to rely on advertisement revenues. Third, half of paid non-game apps are ad free, whereas few of paid game apps are ad free. Thus, game apps tend to combine both revenue sources when they charge consumers. Fourth, the business model share of some non-game categories including comics and social apps are similar to game categories. The average usage time of these categories are higher than other non-game categories and the demand structure may be similar to game apps.

[18]See `https://support.google.com/googleplay/android-developer/answer/9859673`, accessed on February 13, 2021.



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

Figure 3: Summary statistics at market-level

5.3 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 operate in Japan, such as AdMob, *AdColoy*, *InMobi* and *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 units, 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.

5.4 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.[19]

5.5 Numerical Representation of Product Description

Morphological analysis To use product descriptions data in demand estimation, we first convert the sentences in the product description into a bag of words by parts of speech. In Japanese, because words are not separated in a sentence, we employ a morphological analysis engine to split Japanese sentences into a bag of words and classify words into parts of speech. 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). MeCab also distinguishes the parts of speech. We made a bag of words for each parts of speech that is converted into multi-dimensional vectors using the methods described in Section 5.5.

Distributional Hypothesis and Word Embedding We use *nwjc2vec*, developed by Asahara (2018), to numerically represent in-text product descriptions. The 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.

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 multi-dimensional numerical vectors of real numbers that well approximate the meanings of words represented in distributional statistics. The resulting multi-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

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

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 latter 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."

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} is 8, the number of negative samples is 25, and the range of character lengths of n-grams is 3 – 6.

Converting word vectors into a product characteristics Stopwords are the words that are used too often to differentiate sentences, such as "a" and "the." We use the percentage of apps using a word as criteria to define stopwords. We tried 100%, 95%, 75%, 50%, and 25%. For each part of speech, such as noun and verb, we chose criteria that made the p-value of the joint significance of the high-dimensional features the lowest.

As word vectors are numerical vector of words, we have to convert numerical vectors of words in a product description into the semantic vectors of the product. We use the weighted average of numerical vectors of words in an app as semantic vectors of the product. Inverse frequency of words in each app category are used as weight.

We use nwjc2vec to convert app descriptions described in Section 5.2. 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 the app's Japanese descriptions.
2. Split Japanese descriptions into a bag of words by parts of speech using Mecab.[20]
3. Load nwjc2vec using gensim.models.fasttext.
4. Convert each words into multi-dimensional vectors using gensim.models.fasttext.
5. Estimate low-dimensional specification.

[20] The number of words in our data is 22,118.

6. Define stopwords by parts of speech by using p-value of rigorous lasso.

7. Take a weighted average of the multi-dimensional vectors by parts of speech.

8. Take an average of the multi-dimensional vectors across parts of speech.

6 Estimation Result

Estimates of demand parameters Table 3 shows the estimates of key demand parameters, marginal utility of money α_y, marginal disutility of advertisement α_a, satiation parameter η, and anticipation parameter κ.

The ratio of α_a to α_y is the disutility of digesting 1,000 impressions of advertisement in JPY. Table 4 shows that the disutility amounts to JPY 24.7 for applications and JPY 27.2 for games. The parameter estimates indicate that on average the disutility of watching one unit of advertisements is JPY 24.7 for applications and JPY 27.2 for games. Because the average advertising price is JPY 425.8, these amount to 5.8% and 6.4% of the app's advertising revenue. These numbers are smaller than the estimate of Facebook presented in the study of Benzell and Collis (2020). Based on a survey, they estimated that the disutility of advertising is approximately 20% of Facebook's advertising revenue. There are several explanations for this difference. First, it may be merely an estimation error due to the difference in the approach: survey based or revealed preference based. Second, it may be because we use a nominal unit of advertisement as a basis of calculation. Advertisements are often transacted at the nominal number of impressions, but it is not certain whether a consumer actually pays attention to and is "impressed" by an ad shown. Thus, the disutility of watching advertisement can be estimated to be less based on the revealed preference.

The satiation parameter η is nearly equal to zero for both applications and games, indicating that the mean indirect utility of usage is approximately linear in the usage time. The anticipation parameter κ is 7.94 for applications and 52.5 for games. In a separate analysis, we found that then the model cannot fit to either usage or download data if we force the anticipation parameters κ to be 1. This indicates that consumers anticipate using apps for multiple weeks when they decide download. If we could use a consumer-level panel data, we would have estimated a fully dynamic model to capture this effect. But it is beyond the scope of this paper. This prohibits us from studying dynamic counterfactual scenarios such as the customer base formation and inter-temporal price discrimination.

Table 5 shows the estimates of mean β_d and standard deviation σ of random coefficients of observed characteristics in the download-related utility. When downloading, consumers do not appreciate either positive or negative words. Consumers like a longer description for non-game applications, but not for games. Among non-game applications, entertainment, social, and comic apps are the most popular, and education and news and magazines are the least popular. Among game apps, puzzle and casual games are relatively popular, and action and role playing games are unpopular. The standard deviation of the random coefficients for non-game apps are negligible,

Table 3: Estimation results of demand non-linear parameters

Parameter	Application	Game
α_y	0.0194	0.000856
α_a	0.479	0.0233
η	0.01	0.0105
κ	7.94	52.5

Table 4: Implied advertisement disutility

Application	Game
24.7	27.2

whereas they are relatively large for game apps. This means that game apps are differentiated at the category level and consumers are likely segmented into each category.

Table 6 shows the coefficient of observed characteristics in the usage-related utility, β_u. We do not allow for random coefficients for the usage-related utility. For non-game applications, entertainment, communication, personalization, news and magazines, social, and video players apps have relatively higher usage-related utility. Education, music and audio, tools, photography, and productivity apps have lower usage-related utility. The estimates are reasonable except for music and audio. Games tend to have higher usage-related utility and role playing games has the highest value.

Figure 4 shows the estimates of coefficients on word vectors in descending order with the size of the absolute value of the estimates. We first chose the relevant dimensions of word vectors by applying a rigorous LASSO to regressions of download- and usage-related unobserved fixed effects $\hat{\xi}_{dj}(\hat{\theta})$ and $\hat{\xi}_{uj}(\hat{\theta})$ estimated from the low-dimensional model on word vectors. This procedure selected 41 dimensions for download and 29 dimensions for usage. The joint significance test of the relevance of these dimensions are rejected at 1% level. We then included these dimensions as additional observed characteristics in the estimation of the high-dimensional model.

Table 7 shows the estimates of marginal cost of usage λ and the slackness parameter χ. The slackness parameter χ are close to zero for both non-game and game apps. This means that the implicit optimal price is close to zero for free apps and apps business model can flexibly respond to the changes in the market environment. This may be because we identified χ by apps that have both free and paid versions. For other free apps, the slackness may be greater, but no data identifies the value unless we impose an distributional assumption on the download-related marginal cost shocks. In the counterfactual simulations, we consider multiple scenarios by changing the value of χ or by imputing the distribution of free apps' marginal cost shocks from that of paid apps' marginal cost shocks.

Table 5: Estimation results of download taste parameters

(a) Application

Parameter	β_d	σ
Constant	-10.7	0.00252
Positive sentiment	-1.49	0.000198
Negative sentiment	-3.25	0.0017
Log of number of characters	1.31	0.0042
Entertainment	1.92	0.000973
Education	-2.84	0.000219
Communication	0.861	9.03e-05
Personalization	-1.69	0.000671
Music and audio	-0.682	0.000758
News and magazines	-3.22	0.00494
Lifestyle	-1.62	0.000145
Social	2.87	0.000469
Video players	0.619	0.00308
Comics	1.82	0.000204
Tools	-1.49	9.44e-05
Photography	-0.328	4.5e-05
Productivity	-1.01	0.00074

(b) Game

Parameter	β_d	σ
Constant	2.33	0.137
Positive sentiment	-2.43	0.468
Negative sentiment	-1.41	0.0972
Log of number of characters	-0.44	1.21
Puzzle	-1.66	3.33
Card	-2.82	0.591
Casual	-1.75	0.0991
Sports	-6.75	0.507
Strategy	-7.14	0.074
Simulation	-3.05	7.95e-05
Action	-21.1	15
Role playing	-26.5	17.2
Casino	-3.64	0.422
Adventure	-3.9	0.237

Table 6: Estimation results of usage taste parameters

(a) Application

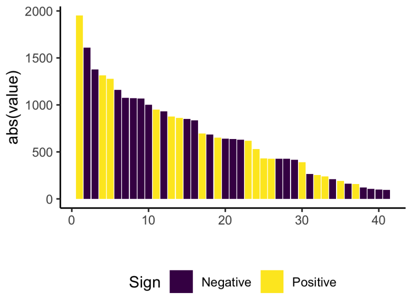
Parameter	β_u
Constant	35
Positive sentiment	-0.00586
Negative sentiment	0.000328
Log of number of characters	-0.000545
Entertainment	0.0103
Education	-0.0121
Communication	0.0124
Personalization	0.00614
Music and audio	-0.011
News and magazines	0.00727
Lifestyle	-0.00557
Social	0.0149
Video players	0.0125
Comics	0.00594
Tools	-0.0131
Photography	-0.0044
Productivity	-0.0152

(b) Game

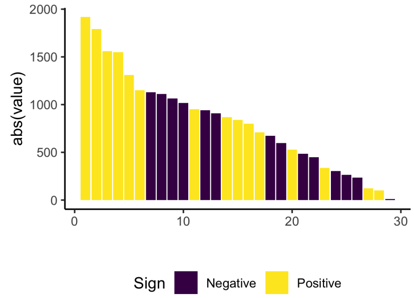
Parameter	β_u
Constant	1.52
Positive sentiment	0.0023
Negative sentiment	-0.0131
Log of number of characters	0.00593
Puzzle	0.00942
Card	0.0148
Casual	0.0108
Sports	0.033
Strategy	0.034
Simulation	0.0188
Action	0.0116
Role playing	0.0342
Casino	0.0126
Adventure	0.0103

Table 7: Estimation results of supply parameters

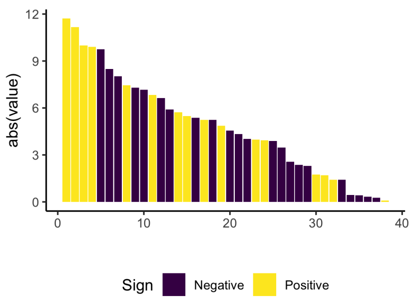
Parameter	Application	Game
λ	77.9	9.57e-06
χ	5.13e-10	0.00632



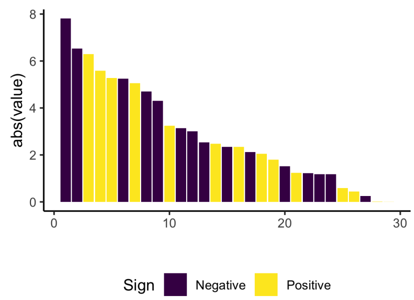
(a) Application: γ_d



(b) Application: γ_u



(c) Game: γ_d



(d) Game: γ_u

Figure 4: Estimation results of word-vector coefficients ordered by the absolute value

7 Relevant Market Definition

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 including 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. 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 SSNIC test.

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.

7.1 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 5.2.

Small but significant and non-transitory increase in cost (SSNIC) tests We introduce a formal definition of SSNIC test that changes the cost to consumers using the competition-in-utility framework in Section 3.4. 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.

A we defined in Section 3.4, the cost c_j of app j is the gap between its maximal mean utility δ_j^0 and its actual mean utility δ_j, that is, $c_j = \delta_j^0 - \delta_j$. Setting the cost c_j leads to the mean utility $\delta_j = \delta_j^0 - c_j$ and gives the per-consumer profit $\bar{\pi}_j(\delta_j^0 - c_j)$ in our competition-in-utility framework.

Let $c = \{c_j\}_{j \in \mathcal{J}}$ be the costs of the apps and $s(c) = \{s_j(c)\}_{j \in \mathcal{J}}$, $q(c) = \{q_j(c)\}_{j \in \mathcal{J}}$, and $\pi(c) = \{\pi_j(c)\}_{j \in \mathcal{J}}$ be the download shares, usages, and profits under c, respectively. Let c^* denote the benchmark equilibrium costs.

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

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

$$\Delta\pi_{\mathcal{M}}^{SSNIC} \equiv \left(\frac{\sum_{l \in \mathcal{M}} [\pi_l(c^{SSNIC}) - \pi_l(c^*)]}{\sum_{l \in \mathcal{M}} \pi_l(c^*)} \right); \tag{46}$$

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

This definition of SSNIC test encompasses the notion of SSNIP test because when there is no advertising, a κ% increase in the costs is equivalent to a κ% increase in the prices. Furthermore, by summarizing the impacts of advertising intensities and prices into uni-dimensional costs, our definition of SSNIC test enables us to compare the apps with different monetization policies.

Order of testing price/cost increase For SSNIC tests, we sequentially add new products to the portfolio of the hypothetical monopolist. The profit change attributes to 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. Moreover, we also use a strategy based on the cosine similarity in the semantic vector space.

7.2 The Equilibrium Relevant Market Cost (ERMC) Test

Finally, we introduce the ERMC test, in which the cost levels are evaluated at the values in the new equilibrium. Consider a hypothetical monopolist that owns apps in set $\mathcal{M}$. Suppose that the ownership of the other apps are the same as in the benchmark. Let δ^{**} be the new equilibrium under this market structure. We define the Equilibrium Relevant Market Cost (ERMC) index as follows:

$$C_{\mathcal{M}}(\delta) \equiv \left(\frac{\sum_{l \in \mathcal{M}} s_l(\delta)\delta_l}{\sum_{l \in \mathcal{M}} s_l(\delta)} \right). \tag{47}$$

Then, we define the change in ERMC index as follows:

$$\Delta \text{ERMC}_{\mathcal{M}} \equiv \frac{C_{\mathcal{M}}(\delta^{**}) - C_{\mathcal{M}}(\delta^{*})}{C_{\mathcal{M}}(\delta^{*})} \tag{48}$$

Then, the ERM relevant market test proceeds in the following way: Let $\mathcal{M} \subset \mathcal{J}$ and consider $\kappa \in (0, 0.1)$. Then $\mathcal{M}$ is the ERM relevant market if and only if $\Delta\text{ERMC}_{\mathcal{M}} > \kappa$.

7.3 Comparing model-based relevant markets with the product category

Illustration with a social 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 social networking app with the largest number of downloads in the category. We pick up a social networking app for illustration, because the merger and acquisition in this category is often under debate.

Figure 5 plots the change in profits of the hypothetical monopolist along the path of the SSNIC tests. In the plots, the x-axis indicates the index of added apps and the y-axis indicates the change in profits when the cost is increased by 5%. The profit of the hypothetical monopolist increases as the number of owned apps increases. The number of apps included in the relevant market based on the elasticity-based order is 45. The number is 29 in the similarity-based order. The number of apps in the relevant markets is greater than the number of apps in the social category apps. This suggest that the app category in Google Play may not be an antitrust relevant market.

SSNIC test of app categories We test whether a product category of Google Play can be regarded as an antitrust relevant market by applying the SSNIC test changing the consumer's cost by 5%. Specifically, we consider a hypothetical monopolist who owns all apps in a category and force it increasing the consumer's cost by 5%. A category is an antitrust relevant market if the profit changes is positive.

Table 8 shows how the hypothetical monopolist's profit changes by this forced increase in the consumer's cost. The profit changes are negative in all categories in non-game apps except for tools. Although the profit change for tools is positive, the change is only by 0.01%. Therefore, product categories in non-game applications are hardly an antitrust relevant market. The profits drop substantially for comics, communication, entertainment and social apps, implying they have non-negligible number of substitutes outside the product category. This makes sense because these apps offer merely different ways of passing time. Comic apps compete for consumers' time with entertainment apps, communication apps compete with social apps, and so forth.

In contrast, some of game app categories are judged antitrust relevant markets. The profit of the hypothetical monopolist increases for action, casino, casual, puzzle, role playing, and simulation games. Because games are yet another ways of passing time, one may expect that they compete each other across categories. The results of the SSNIC test show this is not the case. As we have seen in the estimation results, the standard deviation of random coefficients on categories are relatively large for game apps. Therefore, games are differentiated at the category level and categories can be regarded as an antitrust relevant market.

ERMC test of app categories We also test whether the product categories constitute an antitrust relevant market using a merger simulation. Table 9 show how the average equilibrium consumer's cost increases when a hypothetical monopolist owns the entire apps in each category. A product category is judged as an antitrust relevant market if the average consumer's cost increases more than 5%.

The tests confirm that none of the product category are antitrust relevant markets for non-game apps. This is consistent with the SSNIC test results, except for tools apps. For game apps, action, casual, puzzle, and role playing games are considered to be antitrust relevant markets. The casino and simulation games, which are judged as relevant markets with SSNIC tests, are not relevant market with the ERMC tests, but the consumer's cost of these categories increase by a few percentages.

One difference of ERMC test from SSNIC test is that the non-merged firms strategically respond by increasing costs and amplify the market power of the hypothetical monopolist. Another difference is that the hypothetical monopolist optimally adjust the consumer's cost across owned apps. Because of these differences, the defined relevant market can be either larger or smaller in ERMC test. Nevertheless, in the current exercise, the results between SSNIC and ERMC tests are largely consistent.

8 Welfare Effects of App Mergers

8.1 App Mergers in Category

To examine the effectiveness of the relevant market definition, we examine the welfare effects of mergers involving all the apps in each category. Additionally, the merger exercise allows us to

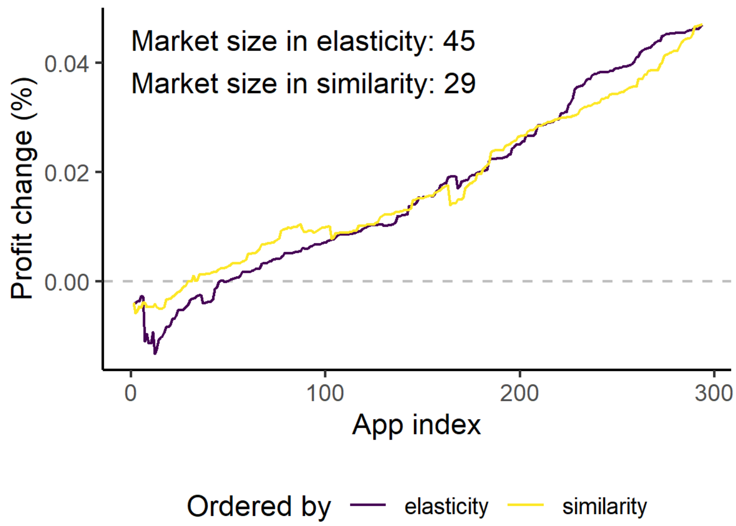


Figure 5: The SSNIC path of the top social app

Table 8: SSNIC test for categories

(a) Application

Category	Profit change (%)
Comics	-6.192
Communication	-12.957
Education	-0.618
Entertainment	-4.131
Lifestyle	-0.105
Music and Audio	-0.168
News and Magazines	-0.438
Personalization	-0.743
Photography	-0.177
Productivity	-0.2
Social	-2.18
Tools	0.01
Video Players	-0.188

(b) Game

Category	Profit change (%)
Action	8.496
Adventure	-0.031
Card	-0.046
Casino	0.103
Casual	0.346
Puzzle	2.944
Role Playing	10.869
Simulation	0.276
Sports	-1.806
Strategy	-0.012

Table 9: ERMC test for categories

(a) Application

Category	Cost change (%)
Comics	0.32292
Communication	0.56706
Education	0.00167
Entertainment	0.14522
Lifestyle	0.0065
Music and Audio	0.5449
News and Magazines	4.02998
Personalization	0.0003
Photography	0.02526
Productivity	0.0138
Social	0.54088
Tools	0.03948
Video Players	0.00348

(b) Game

Category	Cost change (%)
Action	94.23811
Adventure	0
Card	2.65494
Casino	2.94671
Casual	10.59317
Puzzle	139.10911
Role Playing	151.78581
Simulation	4.1362
Sports	0.53669
Strategy	0.86311

uncover the differential effects of mergers to consumers, app developers, and the platform.

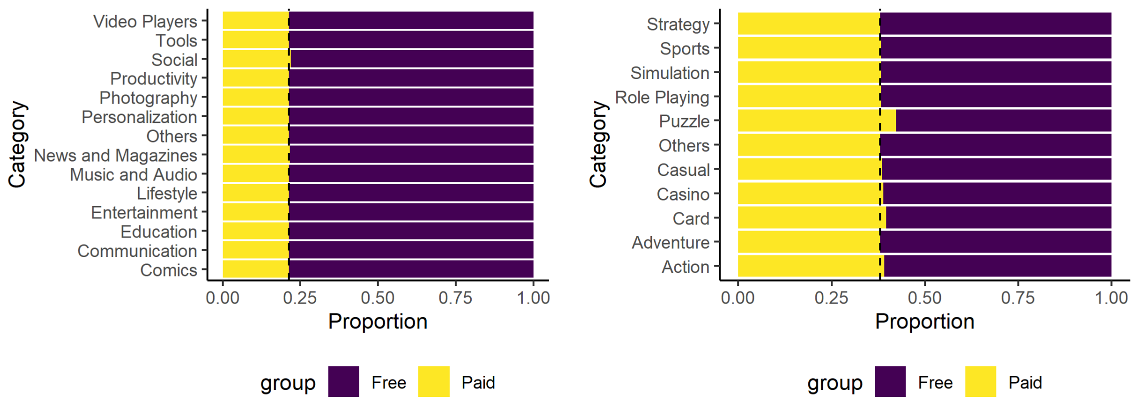
Table 10-(a) shows the welfare effects of mergers involving all apps within each non-game category. For non-game apps, no category-level merger has a substantial effect on consumers, app developers, or the platform. This is consistent with the result of relevant market definition that few product categories of non-game apps are antitrust relevant markets.

Table 10-(b) shows that for game apps, only mergers in product categories that are judged as relevant markets have substantial impacts on welfare. For example, mergers involving action, puzzle, or role-playing apps reduce the total surplus by 1-2.9% and reduce the consumer surplus by 2.9-8.4%. Even if a category is judged as a relevant market, the welfare impact is negligible if the hypothetical monopolist's profit change in the SSNIC test is small. For example, casual game apps are defined as relevant market by both SSNIC and ERMC tests, but its effect on consumer surplus is less than 0.1%. The profit change in this category in the SSNIC test is 0.3% and the cost change in the ERMC test is 10%. The welfare effects of mergers in categories that are not relevant market are negligible.

The results in Table 8 and 10 validate the use of SSNIC tests for predicting the welfare effects of mergers. In Table 8, only action, puzzle, and role-playing apps yielded the increase in the profit of the hypothetical monopolist by more than 1%. Table 10 shows that the welfare effects of mergers are substantial only for those involving action, puzzle, and role-playing apps. Put together, a merger involving a group of apps has substantial welfare effects only if the hypothetical monopolist that owns these apps experience a large increase in profits in the SSNIC test.

In summary, our results suggest that the welfare effects of mergers are aligned with the result of relevant market definition, and that SSNIC test is a useful tool for screening potentially anti-competitive mergers.

We also note that the most of welfare effects of mergers are driven by the increase in prices. Table 11 shows that mergers increases prices but not advertising intensities. This is driven by



(a) Application

(b) Game

Figure 6: Changes in the proportion of paid apps when apps within each category merged

the estimated value of χ being small. When χ is close to zero, then the zero prices of free apps are almost optimal for app developers. Then, the mergers induces app developers to shift their business models from free to paid business models. Figure 6 confirms this result by showing that the proportion of paid apps increases following the influential mergers.

8.2 Top App Mergers

To examine the competitive effects of mergers among dominant apps, we consider the situation in which the developer of the top app in the download ranking acquires the rest of the top apps. We consider the scenario where the top app firm purchases up to top 10, 30, and 50 other apps from original owners.

Table 12 shows the effects of top apps mergers on the consumer surplus and profits of app developers and the platform. For both application and game apps, consumer surplus and total surplus declines, and the profits of app developers increases with mergers, with an increasing magnitude in merger sizes. In particular, the merger involving top-50 non-game apps increase the profit of the app developers by 8% and the profit of the platform by 2%, whereas it reduces consumer surplus by 15.7% and total surplus by 6.6%. For game apps, the merger involving top-50 apps increase the profit of app developers by 35% and the profit of the platform by 24%, whereas it reduces consumer surplus by 6.5% and total surplus by 1.8%.

Compared to the merger involving top non-game apps, the merger among top game apps increases the profit more, and reduces the consumer surplus less. This can be explained by the difference in the levels of taste diversity, captured by the parameter σ. Theoretically, the price effect of mergers increases with the degree of taste diversity. Furthermore, the estimated value of σ in Table 5 shows that there is more taste diversity for game apps than for non-game apps. Therefore, for the game apps, prices increase more rapidly with the merger. Table 13 confirms this observation; average price of game apps increases by 14.4%, whereas that of non-game apps

Table 10: The effects of apps mergers in category: Surpluses relative to the actual

(a) Application

Category	Consumer surplus	Profit app	Profit platform	Total surplus
Comics	1	1	1	1
Communication	0.997	1	0.987	0.997
Education	1	1	1	1
Entertainment	1	1	1	1
Lifestyle	1	1	1	1
Music and Audio	0.999	1	1	1
News and Magazines	0.99	1.01	1.02	0.999
Others	1	1	1	1
Personalization	1	1	1	1
Photography	1	1	1	1
Productivity	1	1	1	1
Social	0.998	1	1	0.999
Tools	1	1	1	1
Video Players	1	1	1	1

(b) Game

Category	Consumer surplus	Profit app	Profit platform	Total surplus
Action	0.971	1.16	1.06	0.99
Adventure	1	1	1	1
Card	1	1	1	1
Casino	1	1	1	1
Casual	0.999	1.01	1.01	1
Others	1	1	1	1
Puzzle	0.95	1.23	1.12	0.98
Role Playing	0.916	1.42	1.24	0.971
Simulation	0.999	1.01	1.01	1
Sports	1	1	0.999	1
Strategy	1	1	1	1

Table 11: The effects of apps mergers in category: Endogenous variables relative to the actual

(a) Application

Category	Price	Ad	Cost	Download	Usage
Comics	1	1	1	0.9999	1
Communication	1.001	1	1.001	0.9986	1
Education	1	1	1	1	1
Entertainment	1	1	1	1	1
Lifestyle	1	1	1	1	1
Music and Audio	1.001	1	1.001	0.9995	1
News and Magazines	1.009	1	1.006	0.9949	1
Others	1	1	1	1	1
Personalization	1	1	1	1	1
Photography	1	1	1	1	1
Productivity	1	1	1	1	1
Social	1.001	1	1.001	0.999	1
Tools	1	1	1	1	1
Video Players	1	1	1	1	1

(b) Game

Category	Price	Ad	Cost	Download	Usage
Action	1.175	1	1.152	0.9933	1
Adventure	1	1	1	1	1
Card	1.003	1	1.003	0.9999	1
Casino	1.002	1	1.002	0.9998	1
Casual	1.005	1	1.004	0.9991	1
Others	1	1	1	1	1
Puzzle	1.198	1	1.172	0.9522	1
Role Playing	1.471	1	1.409	0.9935	1
Simulation	1.007	1	1.006	0.9995	1
Sports	1.001	1	1.001	1	1
Strategy	1	1	1	1	1

Table 12: The effects of top apps mergers: Surpluses relative to the actual

(a) Application

Merger size	Consumer surplus	Profit app	Profit platform	Total surplus
10	0.981	1.01	0.962	0.987
30	0.905	1.05	0.977	0.956
50	0.843	1.08	1.02	0.934

(b) Game

Merger size	Consumer surplus	Profit app	Profit platform	Total surplus
10	0.99	1.06	0.968	0.994
30	0.961	1.21	1.09	0.987
50	0.935	1.35	1.24	0.982

Table 13: The effects of top apps mergers: Endogenous variables relative to the actual

(a) Application

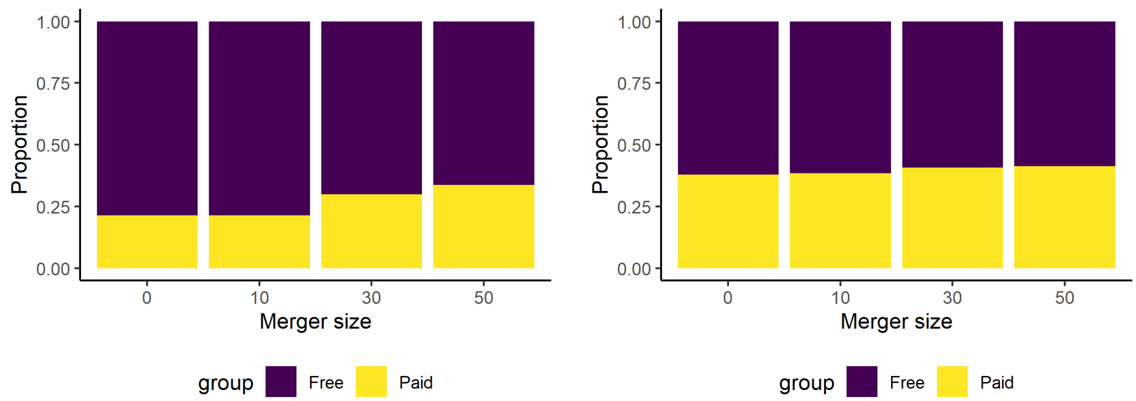
Merger size	Price	Ad	Cost	Download	Usage
10	1.002	1	1.001	0.9906	1
30	1.043	1	1.026	0.9511	1
50	1.054	1	1.032	0.9166	1

(b) Game

Merger size	Price	Ad	Cost	Download	Usage
10	1.014	1	1.012	0.9926	1
30	1.066	1	1.057	0.9739	1
50	1.144	1	1.124	0.9612	1

increases by 5.4%.

Finally, we note the role of endogenous business model in the merger analysis. Table 13 demonstrates that the effect of mergers on the surplus is mostly driven by the increase in prices, because advertising intensities do not change. A part of the price increase can be explained by the increase in the prices of paid apps. At the same time, the free apps shift their business model to paid business models. Figure 7 shows that mergers shifts the business model of free apps to paid apps. Therefore, even mergers involving free apps may have anti-competitive price effects by changing the business models to paid business models. If we evaluated mergers using the models with exogenous business models, we might fail to such an endogenous shifts in business models into account.



(a) Application

(b) Game

Figure 7: Changes in the proportion of paid apps when top apps merged

9 Platform Transaction Fee Reduction

The framework can be used to conduct other types of counterfactual analysis. For example, we can examine the effects of a reduction in the platform fee. Currently, Google Play charges a 30% transaction fee based on the download and in-app purchase revenue and planning to reduce the fee to 15%. Figures 8 and 9 provide the changes in the key variables and welfare measures because the transaction fee is gradually reduced from 30% to 0%. In all the figures, the value of each variable when the transaction fee is 30% is normalized to 1.

In the standard vertical relation where firms pay fees on the sales, a reduction in transaction fees lowers prices, regardless of the form of transaction fee. This is because an increase in per-unit fee increases the marginal cost of the firms and an increase in proportional fee inflates the perceived marginal costs of the firms. Because of the double-marginalization effect, a reduction in fee lowers the price.

Figures 8 and 9 show the opposite happens in the app economy. The median advertisement declines and the median download price increases as the transaction fee declines. The magnitude differs across application and game apps: prices more sharply increase for non-game apps than for game apps, whereas advertising intensities more sharply decrease for game apps than for non-game apps.

Endogenous business model choice by app developers yield this counter-intuitive result. As the transaction fee declines, it becomes more profitable for app developers to collect revenues from download prices. Therefore, some of the ad-sponsored apps shift to paid business model. This leads to an increase in the price and a reduction in the advertisement. Figure 10 shows that the share of free apps declines as the transaction fee drops. There is an additional technical channel. Because the transaction fee takes the form of proportional fee, when the "effective marginal cost", the marginal cost minus advertising revenue, is negative, the presence of proportional fee can reduce the price.

The effects of transaction fees on the profits of apps is straightforward. Because the fee on revenue become high, app developers' profits shrink. The effects of transaction fees on the profit of the platform depends on the value of transaction fees and the estimated parameter values. For game apps the platform's revenue increases with transaction fees, but it is non-monotone for non-game apps. For non-game apps, the prices sharply increases with a reduction in transaction rate, which increases the total revenue and offset the reduction in the share of the revenue that the platform obtains. In contrast, for the game apps, the increase in prices accompanying the reduction in the transaction fee is small and thus not sufficient for compensating the reduction in the revenue share, thereby reducing the profit.

The endogenous business model effect dominates the double-marginalization effect when a non-negligible number of free apps exists and their profitability from pricing and advertising are at the margin. The estimated parameters and the distribution of the marginal cost shocks exhibit this feature reflecting the fact in data that 21% of non-game apps and 27% of game apps has experienced pricing model change during the data period.

The impacts on consumer and total surplus are ambiguous and differ across application and game apps. For non-game apps, a reduction of transaction fee hurts consumer and total welfare. This follows from the fact that a reduction of transaction fee increases the price at a higher rate than it decreases the advertising intensity. In contrast, for game apps, consumer surplus and total surplus are almost flat, implying that transaction fees have small impact on them. Consequently, for the game apps, the changes in transaction fees lead to only a transfer of surplus between app developers and the platform.

The difference in the share of free as-sponsored apps causes this difference between non-game and game apps. Figure 10 shows that the shift from free to paid apps quickly finishes for game apps and the endogenous business model choice effect no longer works. In contrast, the shift continues for non-game apps.

These results suggest that in two-sided markets, the standard result of double-marginalization may not hold and that policymakers should be careful about naively applying the existing results in one-sided markets.

There are several caveats to the interpretation of the impacts of platforms fees. Because our structural model is a static model of competition in prices and advertisements, our framework ignores long-run effects on investments, entry, and any other dynamics, which makes our results about the platforms fees on welfare bias in either a positive or negative direction. On the one hand, with a high transaction fee, few app developers would enter, and app developers would invest less, which would generate the negative welfare effects of platform fees. On the other hand, the presence of platform fees facilitates the platform's investments in market infrastructures such as payments services, SDKs, and recommendation systems, which would generate the positive welfare effects of platform fees. Those two long-run factors are absent in our static framework, which would make the results about the welfare effects of platform fees bias in either a positive or negative direction. In this regard, our result can be interpreted as a benchmark result that would be in effect in the very short run.

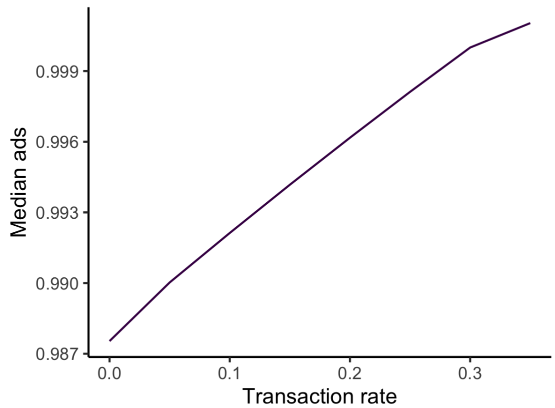
10 Conclusion

In this paper, we demonstrated that many apps combine multiple monetizing policies, charging price and showing advertisement to consumers, and often change their business model using the data set from the Japanese app market. We proposed a novel model of ad-sponsored media with endogenous business model choice. The model allowed app developers to monetize either by charging price or showing advertisements to consumers. The endogenous business model choice was characterized by non-negativity constraints of prices and advertisements. The model defined an equilibrium over consumers' downloads and usage decisions and app developers' pricing and advertising decisions. We transformed the model to an equivalent mean-in-utility model to define the cost for a consumer to use an app regardless of the monetizing policy. This enabled SSNIC test for defining an antitrust relevant market even when free products exist. We estimate the model using mobile app data from Japan from 2015 to 2017.

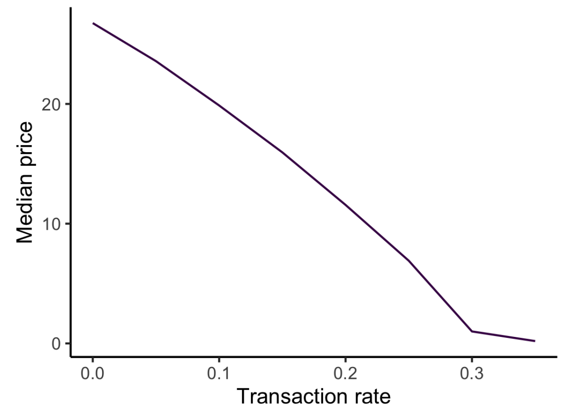
The SSNIC test showed that few non-game categories were relevant markets, whereas so were many game categories. The relevant market definition based on a full equilibrium simulation was aligned with this result. Furthermore, merger simulations showed that the welfare damage was more pronounced in categories where the hypothetical monopolist's profit increased more in the SSNIC test. This result validated the use of SSNIC test as a convenient screening tool.

The analysis of hypothetical mergers and transaction fee reduction confirmed the importance of incorporating the co-existence of multiple monetizing policies and endogenous business model choice of app developers. The changes in price were often achieved by the shift from free to paid apps. Because of this endogenous shift to paid apps, a reduction of transaction fee was predicted to increase the price, rather than decreasing the price, in contrast to the prediction of the standard vertical relation. Without endogenizing the business model choice, we would miss one of the important margin through which mergers and platform's behavior affect the consumer welfare.

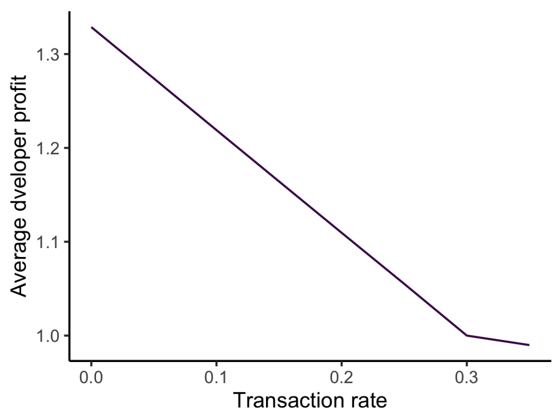
This paper has several limitations. First, we assume that all firms use the ad network as price takers. In reality, some developers would not use the ad network to exert their market power in the ad market. To address this issue, we must directly observe individual advertising intensity and advertising 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 compete with credit cards. Studying the interactions between the mobile app market and the outside market is essential in analyzing the app economy. Fourth, the model is static. The dynamic aspect of the app economy, such as the entry of new apps and the growing dominance of several platforms, can be more important in merger analysis and the analysis of transaction fee reduction. Nevertheless, our model can serve as a static benchmark.



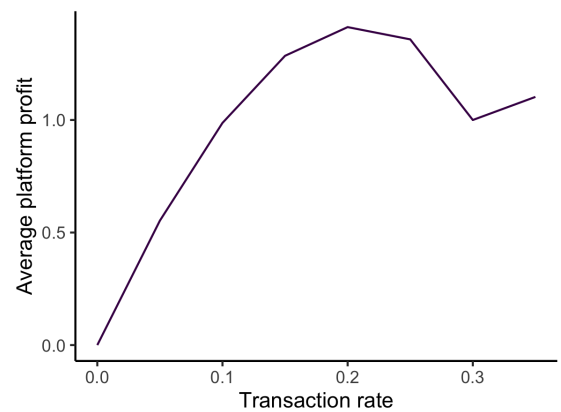
(a) Ad



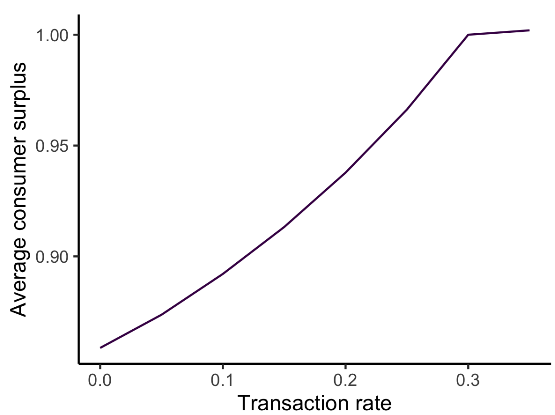
(b) Price



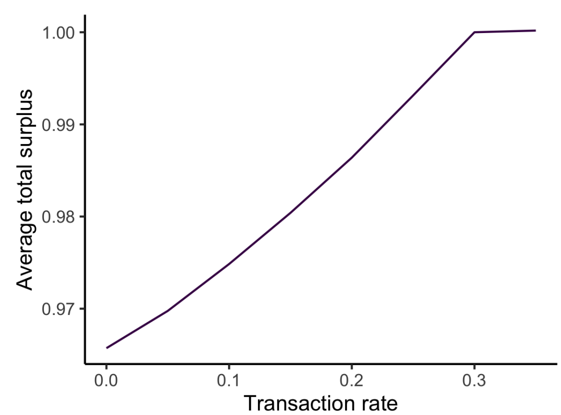
(c) App profit



(d) Platform profit

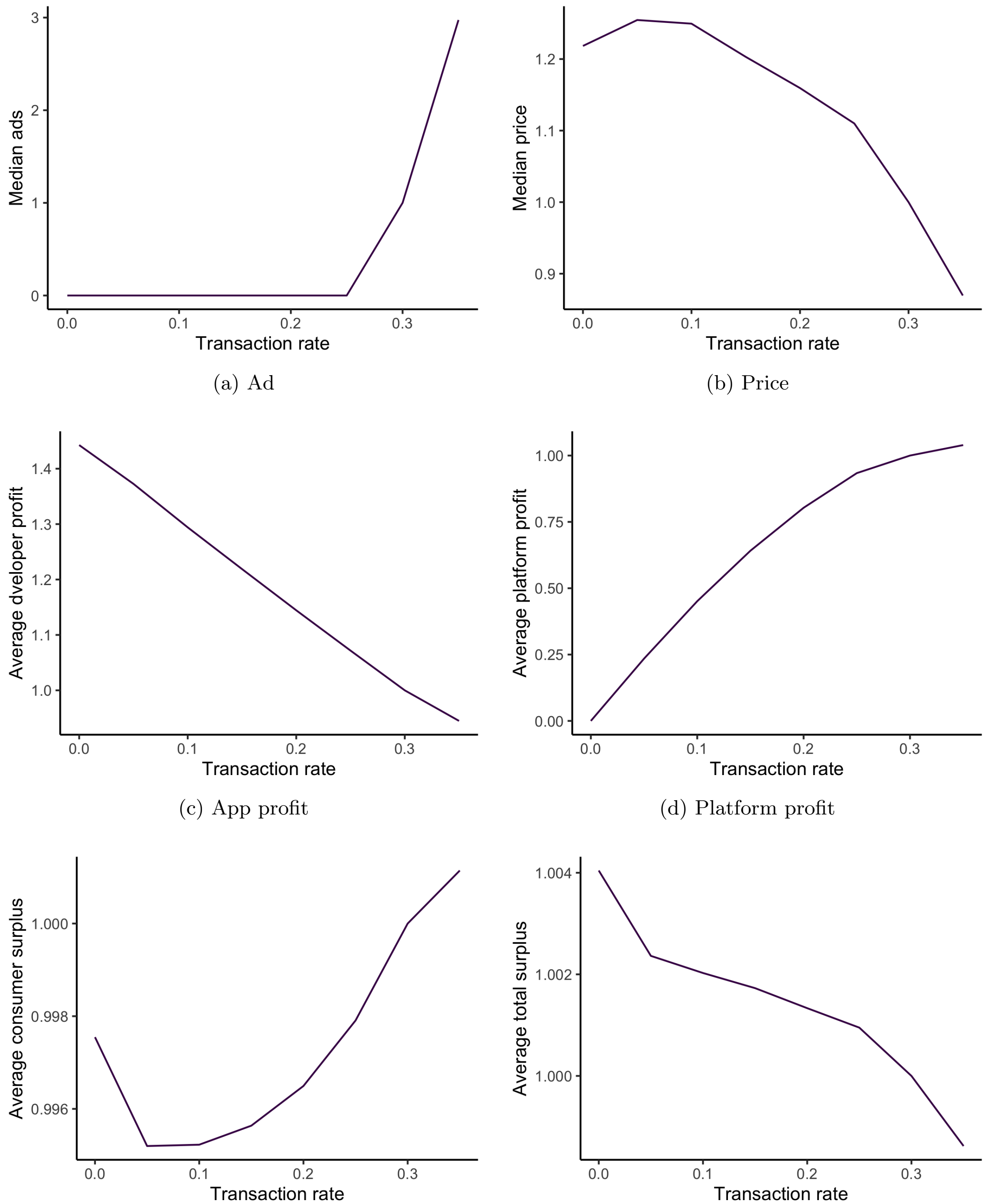


(e) Consumer surplus



(f) Total surplus

Figure 8: The effects of changes in platform fee: Application



(a) Ad

(b) Price

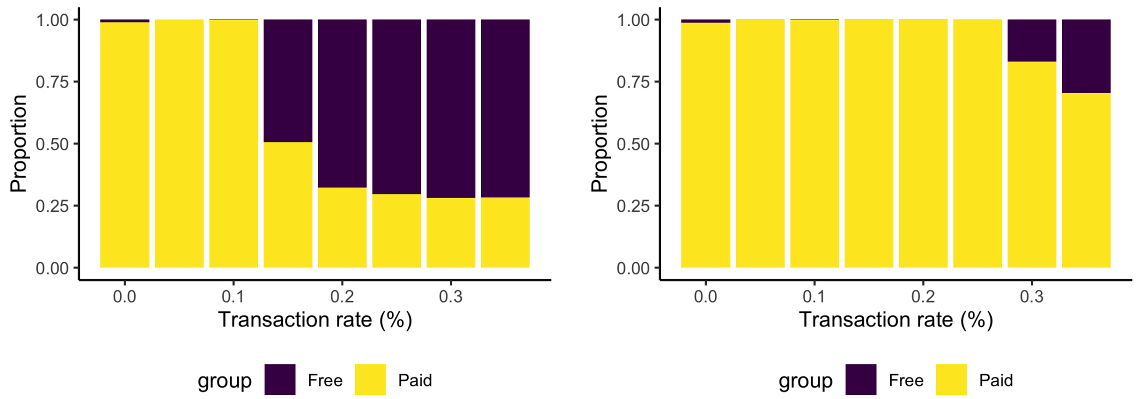
(c) App profit

(d) Platform profit

(e) Consumer surplus

(f) Total surplus

Figure 9: The effects of changes in platform fee: Game



(a) Application (b) Game

Figure 10: Changes in the proportion of paid apps when transaction fee changed

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.

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

Armstrong, Mark and John Vickers, "Competitive Price Discrimination," *The RAND Journal of Economics*, 2001, *32* (4), 579–605.

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.

Belleflamme, Paul and Eric Toulemonde, "Tax Incidence on Competing Two-Sided Platforms," *Journal of Public Economic Theory*, 2018, *20* (1), 9–21.

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

Benzell, Seth and Avinash Collis, "Multi-Sided Platform Strategy, Taxation, and Regulation: A Quantitative Model and Application to Facebook," *SSRN Electronic Journal*, 2020.

Berry, Steven, 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.

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.

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.

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.

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.

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.

Gilbert, Richard J., *Innovation Matters: Competition Policy for the High-Technology Economy.*, Cambridge, MA:, 2020.

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.

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

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.

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.

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.

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.

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

Yuan, Han, "Competing for Time: A Study of Mobile Applications," 2021, p. 41.

A Competition-in-Utility Transformation

We transform the original model of price and advertising competition into a framework of competition-in-utility. A competition-in-utility framework considers a per-consumer profit function $\bar{\pi}_j(\delta_j)$ for each app j that gives mean utility δ_j. The analysis below derives the exact form of the per-consumer profit function $\bar{\pi}_j$

Suppose that a developer d provides mean utility δ_j to consumers who use app $j \in \mathcal{J}_d$. To guarantee that an app j provides utility δ_j, it must satisfy equation (20). Thus, for fixed a_j and δ_j, F_j must satisfy

$$\alpha_y F_j = \max\left\{\frac{\kappa\eta}{2}\tilde{q}_j^2 + \beta_d' X_{dj} + \xi_{dj} - \delta_j, 0\right\}.$$

Let $\bar{F}_j(a_j, \delta_j)$ be defined by the value of F_j that satisfies the above equation. Then, the profit of app j from providing mean utility δ_j and advertisements a_j is given by

$$s_j \hat{\pi}_j(a_j) = s_j\left[(1-\rho)(\bar{F}_j) + \tilde{q}_j(a_j r - \lambda) - \epsilon_j\right].$$

Suppose that $\bar{F}_j > 0$. Then, $\hat{\pi}_j(a_j)$ can be written as

$$\frac{(1-\rho)}{\alpha_y}\frac{\kappa\eta}{2}\tilde{q}_j^2 + \tilde{q}_j(a_j r - \lambda) + \frac{(1-\rho)}{\alpha_y}\left[\beta_d' X_{dj} + \xi_{dj} - \delta_j\right] - \epsilon_j$$

Then, the optimal advertising intensity if characterized by the first-order condition

$$\frac{\partial \tilde{q}_j}{\partial a_j}\left[\frac{(1-\rho)}{\alpha_y}\kappa\eta\tilde{q}_j + a_j r - \lambda\right] + \tilde{q}_j r \leq 0,$$

with equality if $a_j > 0$. Solving this first-order condition, we obtain the value of a_j that maximizes $\bar{\pi}_j$. Let a_j^{int} be such a value, which is independent of δ_j. Thus, as long as $\bar{F}_j(a_j^{int}, \delta_j) > 0$, the optimal advertising intensity is given by a_j^{int}, and $\partial \bar{F}_j / \partial \delta_j = -1/\alpha_y$. The particular value of a_j^{int} is given by solving the above equation.

$$\begin{aligned}
\frac{\partial q_j}{\partial a_j} &= -\frac{\alpha_a}{\eta} \\
q_j &= \frac{\beta_u' X_{uj} - \alpha_a a_j - \alpha_y w + \xi_{uj}}{\eta}.
\end{aligned}$$

Next, consider the case where $\bar{F}(a_j^{int}, \delta_j) = 0$. In this case, we must have

$$\frac{\kappa\eta}{2}\tilde{q}_j^2 + \beta_d' X_{dj} + \xi_{dj} - \delta_j = 0.$$

If there is a solution to this equation in $[0, a_j^{int}]$, such advertising intensity gives the mean utility δ_j. Let $\hat{a}_j(\delta_j)$ be such an advertising intensity. The derivative of $\hat{a}_j$ is given by

$$\hat{a}_j'(\delta_j) = \frac{-1}{\kappa\alpha_a\tilde{q}_j}.$$

Let $\bar{\delta}_j$ be the value of mean utility such that

$$\frac{\kappa\eta}{2}\tilde{q}_j^2 + \beta_d' X_{dj} + \xi_{dj} - \delta_j = 0$$

holds at $a_j = a_j^{int}$ and δ_j^0 be the mean indirect utility of app j at $a_j = F_j = 0$. Then, the app j can provide mean utility $\delta_j \in (-\infty, \delta_j^0]$, and the per-consumer profit function $\bar{\pi}_j(\delta_j)$ is given by

$$\bar{\pi}_j(\delta_j) = \begin{cases} \frac{(1-\rho)}{\alpha_y}\frac{\kappa\eta}{2}(q_j^{int})^2 + q_j^{int}(a_j^{int}r - \lambda) + \frac{(1-\rho)}{\alpha_y}\left[\beta_d' X_{dj} + \xi_{dj} - \delta_j\right] - \epsilon_j & \text{if } \delta_j < \bar{\delta}_j, \\ \hat{q}_j(\hat{a}_j r - \lambda) - \epsilon_j & \text{if } \delta_j \in [\bar{\delta}_j, \delta_j^0], \end{cases} \tag{49}$$

which has the derivative

$$\bar{\pi}_j'(\delta_j) = \begin{cases} -\frac{1-\rho}{\alpha_y} & \text{if } \delta_j < \bar{\delta}_j \\ -\frac{r}{\kappa\alpha_a} + \frac{1}{\kappa\eta\hat{q}_j}\left[\hat{a}_j r - \lambda\right] & \text{if } \delta_j \in [\bar{\delta}_j, \delta_j^0], \end{cases} \tag{50}$$

where the variables with hats ($\hat{q}$) and a superscript int, (q^{int}) are the values of functions evaluated at $\hat{a}_j$ and a_j^{int}. Note that we have $\lim_{\delta_j \searrow \bar{\delta}_j} \bar{\pi}_j'(\delta_j) = -\frac{1-\rho}{\alpha_y}$, which implies that $\bar{\pi}_j$ is smooth at every point in $(\infty, \delta_j^0]$.

Using this transformation, we can reformulate the developer's problem as

$$\max_{\{(\delta_j)\}_{j\in\mathcal{J}_d}} \sum_{j\in\mathcal{J}_d} s_j(\delta)\bar{\pi}_j(\delta_j) \tag{51}$$

$$\text{s. t.} \quad \delta_j \le \delta_j^0, \quad j \in \mathcal{J}_d. \tag{52}$$

The first-order conditions for this problem are:

$$s_j\bar{\pi}_j'(\delta_j) + \sum_{k\in\mathcal{J}_d} \frac{\partial s_k}{\partial \delta_j}\bar{\pi}_k(\delta_k) \le 0, \tag{53}$$

with equality if $\delta_j < \delta_j^0$.