Network Externalities and Cross-Platform App Development in Mobile Platforms

Amandeep Singh, Kartik Hosanagar, Aviv Nevo
The Wharton School, University of Pennsylvania, Philadelphia, PA 19104
{amansin, kartikh, anevo}@wharton.upenn.edu

The benefits consumers get from a mobile phone depend on the software apps available to use on that phone. These indirect network effects are especially important since apps are specific to a platform (e.g., iOS or Android) and are often not available on all platforms at once. We measure the cost to a platform of having delayed entry of some apps, and study the effectiveness of cross-platform development frameworks that allow developers to build an app on one platform and distribute on another. To do so, we develop a model of demand for both handsets and apps that takes into account the heterogeneous indirect network effects between apps and handsets as well as developer entry decisions. We estimate the model using a unique dataset of mobile phone sales and app downloads. We find that delayed developer entry cost the Android platform over $400M in phone sales in 2013 and 2014. We also find that cross-platform frameworks are unable to steer developers towards developing on Android first unless they can generate near-native quality iOS apps.

Key words: Platform Competition; Indirect Network Effects; Online Marketplace; Two-sided Markets.

---

1. Introduction

The value to consumers from, and therefore the demand for, a mobile phone is a function of the software apps available to use on the phone. On the other hand, an app developer's incentives to develop an app to use on a specific type of phone depend on the number of users who adopt compatible phones. This implies that there are indirect network externalities in the adoption of mobile phones, which has important implications for platform competition because mobile platforms are closed and proprietary. Developers have to incur separate development costs to develop apps for each mobile platform. Therefore, many developers initially launch their app on one platform and delay launch on other platforms until they have a better sense of the demand for their app. This sequential app offerings can potentially generate a negative impact on the less preferred platform and therefore it might be in the platform's interest to try to reduce the cost of porting an app from one platform to another. This is especially true if apps tend to be introduced first on one platform. Consistent with this consideration, Google has invested in frameworks like J2ObjC and Flutter to encourage developers to develop apps for Android while retaining the ability to port those apps to iOS.

In this paper, we ask whether such cross-platform development frameworks can help attract developers and mitigate the costs to the platform of delayed developer entry. To answer this question, we develop and estimate models of consumer demand for both handsets and apps, as well a model of developer entry decisions. We estimate the cost of delayed entry by developers on the Android platform and simulate counterfactual app developers' entry scenarios assuming different quality of cross-platform development frameworks.

Our paper contributes to the literature on indirect network effects. In our setting, the network effects are indirect because the value of, for example, an iPhone does not directly depend on the number of other iPhone users since calls can be made to all phones not just iPhones. Instead, having more iPhone users generates incentives for developers to develop more apps and therefore indirectly increases the value of an iPhone. The primary impediment for empirical work in this area has been the lack of rich granular data on both sides of the market. Many papers in the literature have therefore had to rely on various modeling assumptions. For instance, papers on adoption of video cassette recorders (see Ohashi (2003) and Park (2004)) modelled demand as a function of installed base, rather than the number of movie titles available. Similarly many approaches (see Gandal et al. (2000), Nair et al. (2004), Clements (2004), and Corts and Lederman (2009)), approximate the utility for the complementary market as a function of total number of complementary goods without considering which specific complementary goods are available.

To overcome these challenges, we develop and estimate a dynamic structural model of consumer demand for handsets and apps. Two key features of our model are worth highlighting. First, the demand for both handsets and apps is dynamic. Handsets are durable good that consumers use for several years. We therefore follow Gowrisankaran and Rysman (2012) and Lee (2013) and model consumers as forward looking taking into account current and future prices, features, and the "quality" of the apps available on the relevant app store. Similarly, demand for apps is dynamic since once an app is downloaded a consumer is less likely to download it again. Second, the demand for handsets is a function of the quality of the available apps, which is derived directly from the app demand model. This allows us to model, and estimate, heterogeneity in network externalities across apps. Using the size of the complementary market can be a poor approximation when there is significant heterogeneity in network effects. For example, Facebook and Hi5 are both social networking apps, but the unavailability of Facebook on a mobile platform is likely to have a larger effect on mobile sales than the unavailability of Hi5. On the developer side, we develop a model of developer entry and use an inequality based approach developed in Pakes et al. (2015) to estimate the porting cost across the two platforms.

We estimate our model using a novel panel dataset of handset sales and app downloads for both Apple's iOS App Store and Google's Play store. The ability to easily track app launch dates along with daily app downloads and reviews presents a unique research opportunity to quantify indirect network effects. Using this granular data we estimate our dynamic model of demand for handset sales and app download decisions. We use the observed patterns of developers entry to estimate the porting costs across the two platforms.

We find that indirect network externalities are significant in app markets, and on average, increasing the number of apps by 1% leads to an increase in sales of smartphones by 1.5%. Further, we find evidence for significant heterogeneity in externalities exerted by apps. For instance, we find that the externality exerted by the top app is around 2700 and 5600 times that of the median app on iOS and Google Play platform respectively. We observe that the Social Media apps like Facebook have the highest estimated impact on both the platforms. Specifically, removing Facebook from Apple or Android can reduce the demand of respective handsets by -5.9% and -5.6%. Using our model we find that one month access to Facebook app is worth to be around \$13 (or \$9) to Apple (or Android) users.

We find that the Android ecosystem suffers considerable losses due to delayed entry by developers. We estimate losses in revenues from handset sales to be around \$208M in 2013 and \$260M in 2014 for Android. Finally, we evaluate the role that cross-platform development frameworks can play in alleviating such losses for Android. We find that cross-platform development frameworks have a limited effect and are unable to steer developers towards Android. We find that for Android to effectively steer developers to develop for Android at the same time as Apple, they need a cross-platform development framework that can generate a near-native quality app. A framework that is unable to deliver near-native app quality will not be sufficient to convince enough developers to change their entry decisions.

The issues we study in this paper extend beyond mobile apps and are relevant to multiple platform markets such as those for smart TVs (such as Amazon's Fire Stick, Roku TV), emerging home assistance platforms (such as Amazon's Alexa and Google Home), etc. Accordingly, the structural model we develop in this paper has applications to empirical investigations of network effects in those markets as well.

The rest of the paper is organized as follows. In Section 2, we review the relevant literature and position our work within the literature on two sided markets, platform competition, and app markets. In Section 3, we describe the dataset used for the analysis presented in this paper. In Section 4, we lay out the problem formally, and present a dynamic model of demand for consumers decision of handset purchase and app downloads. In Section 5, we present our identification strategy and discuss the specific instruments used for causal identification of indirect network effects. Next, in Section 6, we discuss the specific steps, we take to estimate our structural model. We then discuss the results of our structural model in Section 7. We provide prescriptions and counterfactual analysis in Section 8. Finally, we discuss managerial implications some limitations of our work and conclude in Section 9.

2. Related Literature

There are two streams of work highly relevant to our study. The first is the literature on network effects (direct and indirect). The second is the emerging stream of work on adoption drivers in the mobile app ecosystem. We discuss each stream of work in turn.

Network Effects: Early work (Shapiro and Varian (1998), Farrell and Saloner (1985) and David (1985)) looked at direct network effects and tried to analytically model the impact of network effects on industry formation and evolution. They investigated the social and private incentives to achieve "compatibility" in a single-product network. Recently, there has been interest in two-sided platforms such as credit card networks wherein the benefit to one side (consumer/merchant) is a function of the number of participants on the other side (merchant/consumers). Armstrong (2006), Rochet and Tirole (2003), and Gabszewicz and Wauthy (2004) study the role of network effects on competition between two-sided platforms.

Network effects may also be indirect wherein the value of a product is a function of the usage or adoption of a complementary product. Classic examples are hardware/software complementaries wherein the value of a hardware platform such as a smartphone or a gaming console is a function of the software available on it. Economides and Salop (1992) provide a general framework to study indirect network effects and derive equilibrium pricing under different market structures and levels of competition. Parker and Van Alstyne (2005) explore strategic pricing behavior and product design decisions in markets with strong indirect network effects. They show that firms may find it profitable to discount one product in order to stimulate demand and increase the price of a complementary product. Church et al. (2008) show that indirect network effects can also give rise to adoption externalities. Clements (2004) examines the social and private incentives to achieve standardization and concludes that settings with direct and indirect network effects may lead to divergent outcomes. Hagiu (2006) compares the relative efficiency of a monopoly controlled platform to an open platform and studies how the monopolist can use its power to internalize indirect network externalities. In summary, multiple theoretical studies have shown that if indirect network effects are significant, they impact pricing, competition, and market outcomes.

Not surprisingly, there is burgeoning interest in empirically validating and quantifying the extent to which indirect network effects actually exist in product markets. Early empirical work (Shankar and Bayus (2003), Ohashi (2003),and Park (2004)) model indirect network effects as direct networks effects, assuming that consumer utility for a good increases as a function consumers network size and strength. More recent work (Gandal et al. (2000), Nair et al. (2004), Clements and Ohashi (2005), and Corts and Lederman (2009)) formally model the indirect network effects and show they are quite significant in platform markets. For example, Nair et al. (2004) show that in the market for PDAs, indirect network effects explain roughly 22% of the log-odds ratio of sales with the remaining 78% explained by price and model features. However, all these papers quantify only the aggregate network effect and largely ignore the fact that quality of the complementary good (usually the software) could be highly skewed. Hence, just using the count of the number of software applications available on a hardware platform could be an inaccurate measure of overall utility derived from the complementary goods. In the context of smartphone platforms where there are millions of apps present in app marketplaces, some leading apps may be worth much more than the median app. Recognizing that indirect network effects may not always be homogeneous, Lee (2013) looks at the impact of exclusive titles on video-game industry structure. Unlike video games, mobile apps behave much differently as they have evolving qualities and versions. This warrants us to model changing consumer expectations over evolving app quality. To our best knowledge, this is the only other paper to look at heterogeneous network effects and our paper is the first study to quantify the impact of heterogeneous network effects in the market for mobile handsets and apps.

Recently many studies have looked at estimating the economic value consumers attach to free goods and services like Facebook. For instance, Brynjolfsson et al. (2018) and Brynjolfsson et al. (2019) measures the dollar value consumers are willing to pay for one month access to various free goods like Facebook. However, they primarily rely on survey data for their analysis. Several studies (like Hausman (2012)) have pointed out the limitations of such valuations surveys. For instance, Hausman (2012) points out how respondents in valuations surveys are often not responding out of stable or well-defined preferences and might be inventing their answers on the fly. Using a structural model of demand and purchase decisions made by consumers, we provide an alternate way of measuring dollar value consumers attach to free goods or services.

Further, prior literature has tried to study how converters can help entrants overcome the influence of the incumbent's installed base by enabling cross-technology interoperability. For instance, Choi (1996) and Sen et al. (2010) discuss the effect of converter's efficiency levels on market structure and competition. While these studies investigate the role of converters in platform markets, all of them are theoretical in nature. To best of our knowledge, ours is the first empirical study that investigates the role a converter can play in two-sided marketplaces.

Our paper also relates to broad literature on demand estimation. Prior literature (for instance Heitkötter et al. (2012), Melnikov (2013), Gowrisankaran and Rysman (2012), Conlon (2012), Aguirregabiria and Nevo (2013)) has shown how not incorporating the dynamic nature of goods can lead to biased estimates of price elasticities. Handsets are durable goods that consumers hold on to for multiple years. We follow literature on Gowrisankaran and Rysman (2012) to a dynamic model of demand that accounts for consumers forward looking behavior.

Adoption of mobile apps: There is an emerging literature around adoption drivers of apps and app marketplaces. Ghose and Han (2014) study the demand for apps using a static model of demand. Li et al. (2016) quantifies the effect of buying downloads and evaluate the effect of visibility on app diffusion. Bresnahan et al. (2014) study the platform choice of app developers. Bresnahan et al. (2014) discusses value creation in the highly skewed app marketplaces and provides credence to our approach for modeling app heterogeneity separately. Wang et al. (2018) discusses how the copycat apps affect the demand for original apps. Allon et al. (2019) looks at the developer side and explains how characteristics of platforms review system could provide diverging incentives to app developers. While these studies investigate multiple adoption drivers, none of them investigate or quantify the role of indirect network effects.

3. Data and Descriptive Analysis

In this section we describe our data and provide initial descriptive analysis to motivate the model that follows.

3.1. Data

We use panel data on monthly smartphone sales and daily app downloads in the U.S from January 2013 to December 2014. The smartphone data and the apps data come from two different sources.

The smartphone sales data are from Counterpoint Research, a marketing research firm that tracks smartphones sold in the U.S. We observe the number of smartphones sold and average price of each handset sold every month. We combine these sales data with handset characteristics. After dropping handsets that report either zero sales or price, we are left with data for 17 Apple and 53 Android-based handsets. In Table 1, we display the summary statistics of the characteristics of these handsets. Overall, 40% of the sales observed in our dataset are of Apple handsets, with the rest sales of Android devices. iPhone 5S and Samsung S4 are the highest selling Apple and Android devices, respectively. The average price of Apple handsets is almost double the average price of Android handsets, despite the fact that Android handsets on average have larger displays, more RAM, more storage and a longer battery life.

The app data come from App Annie, a firm that collects data from both Google Play and the Apple app store. We observe the list of top 500 apps downloaded (across all genres) on the Apple and Google Play U.S. store each day 2013-2014, the same two-year period for which we observe

Table 1. Handset Characteristics

	Apple		Android		Overall	
	Mean	Std	Mean	Std	Mean	Std
Price ($)	559	137	305	134	365	172
Display Size (inches)	4.0	0.4	4.6	0.6	4.5	0.7
Primary Camera (Megapixels)	8.0	0	8.0	3.7	8.0	3.2
RAM (Megabytes)	985	135	1526	630	1398	600
Storage (GB)	35	26	18	17	22	22
Battery (milliamp hours mAh)	1594	336	2276	531	2114	571

Note. The summary statistics are computed across 70, 17 Apple and 53 Android, handsets over the 24-month period. Display size refers to the diagonal length of the screen measured in inches. Primary camera refers to resolution of the back camera measured in megapixels. RAM is the random access memory of the handset measured in MB. Storage is the physical capacity available in the handset, measured in GB. Battery is measured in milliamp hours mAh.

handset sales.[1] This process results in $6,950$ and $13,731$ unique Google Play and Apple apps, respectively.[2] For each of these apps, we observe the genre, daily price and sales rank, number and prices of its in-app purchase assortment, daily review activity, app developers' (firm) characteristics, and versioning activity. We convert the daily sales rank into an estimate of the daily downloads for each app by building on methods proposed by Garg and Telang (2013). For details please see Appendix E.

In Table 2, we display the summary statistics of the characteristics of apps in our sample. The paid apps available on Google Play store are priced higher, on average, than Apple iOS store. As we previously noted, the Apple Store has more apps, a fact we need to account for when comparing averages. Indeed, the price difference is driven by the mix of apps: when we compare prices for the same set of paid apps we find that the difference between the stores is not statistically significant(i.e., \$2.81 versus \$3.09 (p-value = 0.15) on Apple vs Google Play store, respectively). We find that customers on Android leave many more reviews (and higher ratings) than on Apple iOS. This differences persists even when we hold the set of apps constant across the two app stores (1331 versus 3608 monthly reviews (p-value < 0.01) and 3.89 versus 3.96 average ratings (p$-$value = 0.03) on Apple vs Google Play store, respectively). This might be driven by the subtle differences

[1] The data for Android apps are only from Google Play and not other app marketplaces. The same is true for Apple, but unlike the Android ecosystem, there are no other apps stores on Apple devices. This is not a major limitation given that Google Play has the lion's share of the Android app market in the U.S. We focus on the top 500 apps downloaded each day since we believe that apps downloaded less frequently are not likely to have significant impact on handset demand.

[2] One potential reason for sampling more unique apps from Apple app store compared to Google Play, could be more developers choosing to launch their apps on Apple first. Since developers not able to profitably operate their apps on Apple might not choose to launch on Google Play later, it could create a potential imbalance across the two app stores.

in their review system policy: Google Play displays apps' reviews over entire history (i.e., across versions) whereas Apple restarts apps' reviews to cover only their most recent version released.

We also find that Apple developers tend to iterate over versions quicker than Android for even the same app (2.21 versus 4.34 months (p-value < 0.01)). This could reflect the more demanding and competitive nature of Apple marketplace. Finally, we also observe that developers on Google Play tend to maintain bigger portfolios than Apple developers.

These subtle differences in app characteristics across the platforms can create varying levels of quality perceptions, even among the same apps on Apple iOS and Google Play. This difference in quality levels, even among the same apps serves as a source of variation for identifying apps' externalities on handset demand.

Table 2. App Characteristics.

	iOS		Google Play		Overall	
	Mean	Std	Mean	Std	Mean	Std
Price	3.14	4.30	3.91	3.15	3.40	5.35
Downloads (Daily)	353	1425	259	1292	326	1389
# of Monthly Reviews	783	3018	2091	5799	1234	4235
Average Customer Review Rating (Quality)	3.95	0.70	4.03	0.50	3.97	0.63
Average Version Age (months)	2.20	2.90	4.50	6.01	2.98	4.36
# of Versions	12	10	13	15	13	12
Firm Size	17	32	25	47	20	38

Note. The summary statistics are computed across our entire sample of 13,731 ios and 6,950 Google Play apps over the 2-year period. For price we only report statistics for paid mobile apps. # of Monthly Reviews refer to the volume of customers reviews received in the focal month. # of Versions refer to the number of versions released for each app as observed in December 2014. Average Customer Review Rating refers to the average valence (out of 5) of customer reviews received by the app in the focal month. Firm Size refers to the portfolio size of the apps' developers in our sample. 59% of Apple apps have an In-App Purchase option vs 56% for Google Play apps.

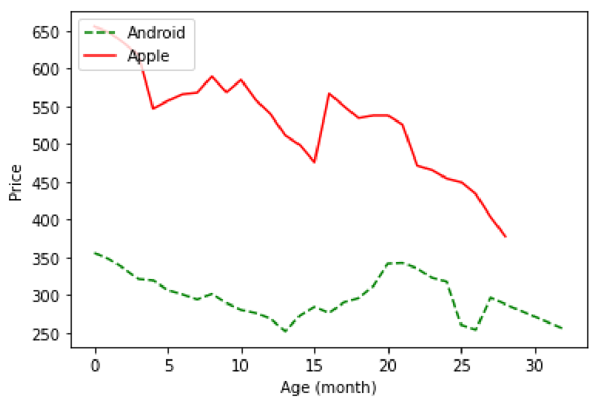
3.2. Evolution over time

We now turn to the evolution of characteristics over time, which will motivate our modeling approach below. In Figure 3 we present the trends in handset and app prices, and app quality as a function of age. As we can see in panel (a) retailers tend to discount handsets as they age. This pattern is found in many durable products and is usually attributed to intertemporal price discrimination: retailers set a higher price early to extract surplus from the high-valuation consumers and then discount their prices to attract the lower-valuation consumers.[3] Such pricing behavior by

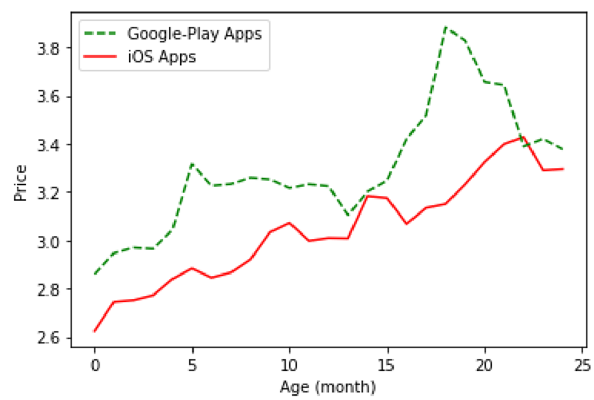
[3] Apple never puts explicit discounts on its handsets, until the launch of a newer version. However, the average sales price of an Apple handset can still show a slight downward pattern due to the various promotions run by retailers.

retailers generates incentives for customers to time their purchases and wait for future discounts. In panel (b) we show a slightly reverse pattern in apps' pricing: apps' prices slightly increase as they age. In Panel (c) we show that app quality, as measured by the app's average rating received in the focal month, shows an upward trend as the apps age. This indicates that as the apps mature their quality levels (as perceived by customers) improve on average.

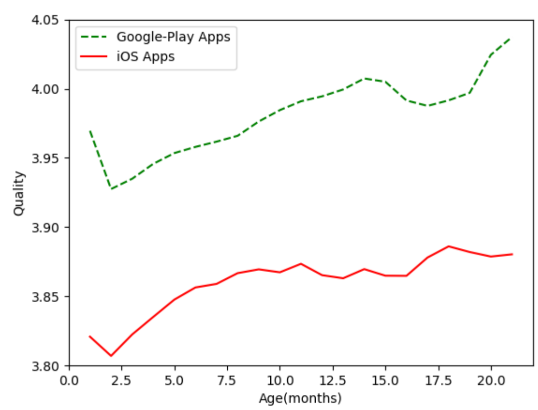
Figure 3. Handset and App Characteristics over Time



(a) Handset Prices



(b) App Prices

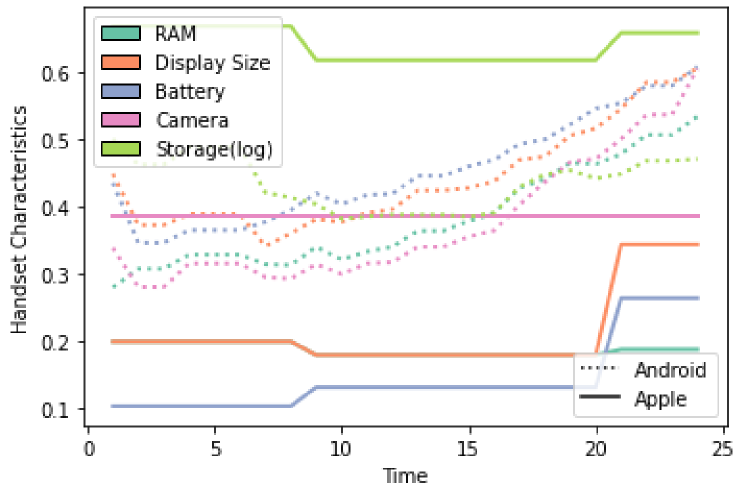


(c) App Ratings

Note: The figures present the evolution of handset and smartphone prices and quality (average customer rating) as they age. In Panel (a) we present the evolution of smartphone prices as they age. The average is calculated over the mix of all handsets of a specific age. In panel (b) we present the evolution of apps' prices (only paid apps) as they age. For this purpose we only average over apps that were launched in and after January 2013. In panel (c) we present the evolution of app quality measured by the average customer ratings they received in the focal month as they mature.

Counterpoint Research takes into account these retail promotions to calculate the average selling price of Apple handsets.

Figure 4. Evolution of Handset Characteristics.



Note. The figures report how the handset features for the entire mix of handsets in the market evolve over time. The solid line denotes the average value of Apple handset characteristics over time and the dashed line indicate the average value of Android handset characteristics over time. For the sake of convenience, each handset characteristic is normalized to a value between 0 and 1. Time period 1 denotes January 2013 and Time period 24 denotes December 2014.

In Figure 4 we examine how handset characteristics (camera, display size, battery, ram, storage) are evolving over time across Apple and Android. Overall, we find that the industry has been steadily improving in each respective handset feature. As with prices, customers can also time their purchases in anticipation of better handset characteristics. As we discuss in the next section, static demand models, which do not account for such anticipatory consumer behavior, may lead to biased estimates of demand elasticities.

3.3. Overlap in Apps offering

In this section, we elaborate on the decisions made by developers that have chosen to offer their apps across both the platforms. Due to the interconnected nature of smartphone and app demand, decisions made by developers affect consumer choices. In our sample, 2103 apps launched on both Apple and Google Play (as observed up to two years after the period of our study, i.e., up until Jan 2017).

In Table 5, we report entry decisions for apps with a presence on both platforms in our sample. We find developers have strong preference for the Apple platform. For instance, the majority (64.6%) of app developers prefer deploying an app on iOS first vs. Android. Specifically, we find that apps are launched on the Apple App store an average 6 months before Google Play. This could affect

Table 5. Developer Entry Decisions

Overall Statistics

Apple First	64.6%
Google Play First	12.2%
Both	23.2%

Statistics by Genre

Genre	Days
Entertainment	196
Games	151
Music	305
Photo and Video	292
Social Networking	238
Utilities	103
Others	232

Note: In this table we report summary statistics for entry decisions of app developers. "Apple First" denotes the share of app developers that chose to launch on iOS first. "Google Play First" denotes the share of app developers that chose to launch on Google Play first. "Both" denotes the share of app developers that launched on both platforms at the same time. We also report the difference between Android and Apple launch dates (in days) across various genres.

the Android platform's attractiveness to customers and could be the reason for concerted efforts[4] carried out by Android to attract developers towards its platform. Across the various genres we find the gap between Apple and Android launch is the longest for apps requiring more development resources (for instance music, photo and video and social networking) and the least for generic apps (like utilities).

4. Model

In this section we introduce a model that allows us to study the impact of indirect network externalities on mobile platforms. On the demand side, we model the consumer's decision to buy a handset and the decision to download an app form the App Store. As we saw in the previous section, price and attributes of handsets and apps are evolving, generating incentives for consumers to time their choices. We therefore use a dynamic model to capture consumer demand for both handset and apps. On the supply side, we model the developers decisions to offer apps on the two platforms.

[4] See `https://medium.com/jay-tillu/flutter-googles-most-ambitious-framework-e3b36ca7091f` and `https://stablekernel.com/article/the-business-case-for-google-flutter/` for more discussion.

4.1. Demand for Handsets

We first consider a consumer's handset purchase decision. Handsets are durable goods that are bought roughly every 3 years.[5] Consumers, in general, can time purchases and indeed faced with declining prices for a given model (Figure 3) and improving attributes across models (Figure 4) they have incentives to do so. For instance, price-sensitive consumers may wait for the price of handsets to decline. Similarly, the less price sensitive consumers may purchase early and leave the market, making the mix of consumers in later periods more price-sensitive. Therefore, to consistently estimate demand elasticises we need to account for this dynamic behavior.[6] In Section 6, we compare the results from a dynamic model of consumer demand to a static model and find, how not accounting for the dynamic behavior of customers can lead to misleading estimates.

In each period t (month) the consumer decides whether to buy one of H_t handsets available at time t or wait until next period. If consumer i purchases handset $j \in H_t$ at time t they get utility

$$u_{i,j,t} = \underbrace{\boldsymbol{\alpha}_i^x \vec{x}_{j,t} + \alpha_i^{p,hs} \log p_{j,t} + \alpha^{\Gamma} \Gamma_{g(j),t}(\alpha_i^{p,app}, \boldsymbol{\alpha}_i^w) + \xi_{j,t}}_{\delta_{i,j,t}} + \epsilon_{i,j,t}^{(hs)} \tag{1}$$

where, $\vec{x}_{j,t}$ refers to the observable characteristics of the handset at t, including camera resolution, display size, ram, storage, and handset fixed effects. $\boldsymbol{\alpha}_i^x$ denotes consumer i's tastes for these characteristics; $\alpha_i^{p,hs}$ captures consumer i's (handset) price sensitivity; $\xi_{j,t}$ captures (time varying) characteristics of handset j that are unobserved by the econometrician but are valued, and observed, by the consumers. Unobserved characteristics, for instance could include unobserved promotional activities, unobserved handset equity, or systematic demand shocks. Some components of the unobserved characteristics are fixed over time and can be captured using dummy variables. Therefore, we model $\xi_{j,t} = \bar{\xi}_j + \Delta\xi_{j,t}$. The error term $\Delta\xi_{j,t}$ is the time varying component of the unobserved attributes. The term $\Gamma_{g(j),t}(\cdot)$ captures the utility the consumer gets from the apps (software) available at the app store $g(j)$. This term varies between Apple and Android phones. For now, we will assume that it is known, and in the next section discuss how we model it. Finally, $\epsilon_{i,j,t}^{(hs)}$ captures random shocks that are assumed to be independently and identically distributed according to the type I extreme value distribution.

We assume that if the consumer buys a handset they exit the market for handsets. Therefore, $u_{i,j,t}$ can be viewed as the lifetime value from purchase of product j. As we noted above, on average consumers replace their handset roughly every 3 years. We observe two years of data and therefore, modeling repeat purchases is not a first order concern (or clearly identified using our

[5] See `https://www.itsworthmore.com/blog/post/years-a-new-smartphone-last`.

[6] See Melnikov (2013), Gowrisankaran and Rysman (2012), Conlon (2012), Aguirregabiria and Nevo (2013) for further discussion.

data). Furthermore, such a simplification significantly reduces the complexity of the problem and makes identification of primary effects of interest (i.e., network effects) easier. We also assume that there is no resale of handsets.

Consumers have the option of not purchasing in period t, in which case they can consider purchasing in the future. A consumer who does not purchase a new handset at time t, i.e. chooses the outside option (denoted as $j=0$), gets the one period utility

$$u_{i,0,t} = \epsilon_{i,0,t}^{(hs)} \tag{2}$$

where $\epsilon_{i,0,t}^{(hs)}$ is assumed to be independently and identically distributed according to the type I extreme value distribution. In principle, consumers might own an older handset. This will in part be captured by $\epsilon_{i,0,t}^{(hs)}$.

At time t, the consumer is faced with $|H_t|+1$ choices, where H_t denotes the set of handsets available at time t. The consumer chooses the option that gives her the maximum expected value, given her information set at t. At the time of choice, consumers know prices and attributes (including those unobserved by us) of all products. They also observe all (current) $\epsilon_{i,j,t}$. Let Ω_t be the state variable that captures all relevant information used by consumers to make their decision. This state will include the set of handsets, their prices, attributes, and associated app store utilities ($\Gamma_{g(j),t}(\cdot)$), and will change over time. Consumers form expectations about the evolution of these variables, as we discuss below, but do not know future values $\vec{\epsilon_i}$.

The Bellman equation for the consumer's problem can be written as

$$V(\Omega_{it}, \epsilon_{it}) = \max_{A_t \in (0,1)} \begin{cases} \epsilon_{i0t} + \beta\, \mathbb{E}(V(\Omega_{it+1})|\Omega_{it}), & A_t = 0 \text{ if no purchase} \\ \max_{j \in H_t}\{u_{ijt}\}, & A_t = 1 \text{ if buys a handset} \end{cases} \tag{3}$$

where, $V(\Omega_{it}) = \int V(\Omega_{it}, \epsilon_{it}) dF_\epsilon(\epsilon_{it})$ is the integrated value function.

The dimension of the state space in the above problem is very high, and therefore the problem is impractical to solve (and take to the data). We therefore follow the literature to reduce the dimensions of the problem in two steps.

First, following Rust (1994), we integrate the logit error ϵ and rewrite (3) using the integrated value function

$$V(\Omega_t) = \ln(\exp(\beta\, \mathbb{E}(V(\Omega_{t+1})|\Omega_t)) + \exp(\zeta(\Omega_t))) \tag{4}$$

where $\zeta_i(\Omega_t) = \ln(\sum_{j \in H_t} \exp \delta_{ijt}))$ is the so-called inclusive value;[7] and $\delta_{ijt} = \boldsymbol{\alpha}_i^x \vec{x}_{j,t} + \alpha_i^{p,hs} p_{j,t} + \alpha^\Gamma \Gamma_{g(j),t}(\alpha_i^{p,app}, \boldsymbol{\alpha}_i^w) + \xi_{j,t}$. As in Rust (1994) we can now solve for the integrated value function, which is only a function of Ω. In other words, by focusing on the integrated value function we significantly reduce the dimension of the state space.

[7] We omit the Euler constant as it does not affect the decision-making process.

Furthermore, as in Hendel and Nevo (2006), Melnikov (2013), and Gowrisankaran and Rysman (2012) we assume that

$$G_i(\Omega_{t+1}|\Omega_t) = G_i(\zeta_{i,t+1}(\Omega_{t+1})|\zeta_{i,t}(\Omega_t)) \tag{5}$$

where $G()$ is the (exogenous) transition process of the state variables. As Hendel and Nevo (2006) note, $\zeta(\Omega)$ can be viewed as a price index. Therefore, this assumptions means that the consumer does not need to know, or keep track, of Ω_t. Nor does the consumer need to predict the future Ω_{t+1}. Instead the consumer just need to keep track of the lower dimension $\zeta(\Omega)$.

This assumption implies that in order to predict the distribution distribution of future states consumers only use the lower dimensional price index ζ_t and not the higher dimensional Ω_t. This can be taken as a statement about how the actual process in which Ω_t evolves or as a behavioral statement about how consumers form expectations. The key is that consumers only track the current value of ζ_t to predict the future distribution of ζ_{t+1} instead of the whole information set Ω_t. This reduces the dimension of the state space to a more tractable single dimension and makes the computation of V numerically feasible. Specifically, it allows us to rewrite equation (4) as follows

$$V(\zeta(\Omega_t)) = \ln(\exp(\beta\,\mathbb{E}(V(\zeta(\Omega_{t+1}))|\zeta(\Omega_t))) + \exp(\zeta(\Omega_t))) \tag{6}$$

We further assume an AR(1) specification for $\mathbb{E}[\zeta_{i,t+1}|\zeta_{i,t}]$ as follows

$$\zeta_{i,t+1} = \phi_{i1} + \phi_{i2}\zeta_{i,t} + \mu_{it+1} \tag{7}$$

where, ϕ_{i1} and ϕ_{i2} are parameters, and μ_{t+1} is identically and independently distributed with mean 0.

Using the model we can construct an expression for the probability of buying each handset. The probability of consumer i purchasing a handset j at time t can be written as the probability of a purchase happening at time t times the probability of purchasing handset j conditional on a purchase happening, which is given by

$$s^{hs}_{ijt} = \frac{\exp(\zeta_{i,j,t})}{\exp(\zeta_{i,j,t}) + \exp(\beta\,\mathbb{E}(V(\zeta_{i,j,t+1})|\zeta_{i,j,t})))} * \frac{\exp(\delta_{i,j,t})}{\exp(\zeta_{i,j,t})} \tag{8}$$

The aggregate market share of handset j in market t, which will be used in teh estimation, can computed by integrating over the distribution of consumers

$$s^{hs}_{jt} = \int s^{hs}_{ijt} d\mathcal{P}^*$$

where $d\mathcal{P}^*$ denotes the distribution of consumer preferences. As we discuss below, this market share can be mapped to the observed market shares of handsets to estimate the model parameters.

4.2. Demand for Apps

Once a consumer purchases a handset, depending on the operating system, they get access to either the Apple app store or Google play store.[8] In the previous section we denoted the benefits from the relevant app store, $g(j)$, by $\Gamma_{g(j)}(\cdot)$. We now explain how we model, and later estimate, these benefits.

At a high level, we need to deal with two main issues. First, consumers typically download multiple apps in a given period.[9] We deal with this by treating each app download as an independent decision.

Second, we need to deal with the fact that apps are durable products: once a consumers downloads an app they are not going to download it again. Ignoring this can lead to mistakes in inference. Consider the following example. suppose there are two apps A and B. Assume that A is launched in period t and is valued more by consumers than B, which is released in period $t+1$. In period t, assume almost all consumers download A. Now, in period $t+1$, there will be very few consumers left that would have not downloaded A; however, the entire market has the option to download B. Thus it is possible to observe greater number of downloads for B than for A. If we ignore this dynamic, and simply use the relative market shares each period to infer the value, we would get biased estimates of customer preferences. We deal with this issue by modeling the consumer's dynamic decision.

We assume that consumer i, compares the perceived lifetime utility, namely, the expected utility over all future periods, for every app l with their reservation utility (and continuation value over future periods) of not downloading the app. If the consumer perceives the (lifetime) utility of an app to be greater than the reservation utility, they download it, otherwise, they wait. We assume that a consumer can download an app only once because the data only accounts for the first download of an app by a user. The (lifetime) utility a consumer i gets by downloading an app l at time t (day) on platform $g(j)$ is

$$u^{app}_{i,g(j),l,t} = \underbrace{\boldsymbol{\alpha}^{w}_{i}\vec{w}_{g(j),l,t} + \alpha^{p,app}_{i} p_{g(j),l,t} + \eta_{g(j),l,t}}_{\psi_{i,g(j),l,t}} + \epsilon^{(app)}_{i,g(j),l,t} \tag{9}$$

where, $\vec{w}_{j,t}$ refers to the observable characteristics of the app at time t; $\boldsymbol{\alpha}^{w}_{i}$ denotes consumer i's taste towards these characteristics; $\alpha^{p,app}_{i}$ captures i's (app) price sensitivity; $\eta_{g(j),l,t}$ captures (time

[8] Some Android manufacturers like Samsung provide their users with access to an exclusive set of apps through their private app stores. The fixed part of these effects gets absorbed into the handset specific fixed effects. The time varying components is captured by the unobservable handset characteristic $\xi_{j,t}$.

[9] Consumers who choose to download apps, download on average approximately 4 apps per month. See, for example, https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/The-2017-US-Mobile-App-Report?cs_edgescape_cc=US)

varying) characteristics of app l that are unobserved by the econometrician but are valued, and observed, by the consumers. As in the demand model for handsets, the Unobserved characteristics, could include unobserved promotional shocks, app fixed effects, or systemic shocks in app demand. Some of these components will be fixed over time and therefore we model $\eta_{g(j),l,t} = \bar{\eta}_{g(j),l} + \Delta\eta^{app}_{g(j),l,t}$. The error term $\Delta\eta^{app}_{g(j),l,t}$ is the time varying component of the unobserved attributes. Finally, $\epsilon^{(app)}_{i,g(j),l,t}$ captures random shocks that are assumed to be independently and identically distributed according to the type I extreme value distribution.

The observable characteristics include the app's current version, current version's age, current period's quality (rating), firm (app developer) size, in-app purchases, and time invariant attributes. App developers continuously update their apps. Apps current version number captures this development and is a good measure of apps maturity. To capture the decay in quality we use version age. Similarly, to control for app's quality we use the consumers ratings in the focal period as a proxy for app's current quality. Finally, the app fixed effects capture any attributes, observed and unobserved by us, that are constant with and app over times and capture perceived quality across apps.

If the consumer decide not to download an app at time t they get the one period utility

$$u_{i,0,t} = \epsilon^{(app)}_{i,0,t} \tag{10}$$

where $\epsilon^{(app)}_{i,0,t}$ is assumed to be independently and identically distributed according to the type I extreme value distribution.

We assume that consumers make the decision of to download an app independently of other apps. So, for example, a consumer's decision of whether and when to download Spotify is independent from her decision on whether and when to download Facebook. As we will see, this assumption makes the model more tractable, and allows us to deal with dynamics. However, it does not account for substitution across apps. Unlike the handset market, where consumers exit after buying a handset, allowing for both substitution and consumer dynamics in the apps market is computationally difficult, as it involves keeping track of all consumption sets over the entire set of apps. Lee (2013) shows that in markets with many products, the loss in predictive power of a model based on independence assumptions is minimal for most products and almost negligible for top products. Since we only include top apps in our analysis, and top apps (for instance Facebook, Instagram, etc) in our context are highly differentiated, this assumption seems reasonable for our application and should not have a significant impact on our analysis. We discuss the robustness of our results to this assumption in Appendix D.

As noted earlier, the consumer decides each period if to download at app or wait. If they wait they have the option to download the app in the future, which makes this a dynamic (stopping) problem. Let Ω_t^{app} be the state variables that captures all relevant information used by consumers to make their decisions. The Bellman equation can be written as

$$W_{i,g(j),l}(\Omega_{i,l,t}^{app}, \epsilon_{i,g(j),l,t}) = \max_{a_t \in \{0,1\}} \begin{cases} u_{i,g(j),l,t}^{app}, & a_t = 0 \text{ if downloads} \\ \epsilon_{i,g(j),0,t} + \beta E[W_i(\Omega_{i,l,t+1}^{app}) | \Omega_{i,l,t}^{app}], & a_t = 1 \text{ if waits} \end{cases} \tag{11}$$

where, $W(\Omega_t^{app}) = \int W(\Omega_t^{app}, \epsilon_t) dF_{\epsilon_t}(\epsilon_t)$.

As in we did in the demand for handsets, we follow Rust (1994) and integrate out the logit error ϵ, to rewrite equation (11) as

$$W(\Omega_t) = \ln\left(\exp\left(\beta\,\mathbb{E}(W(\Omega_{t+1})|\Omega_t)\right) + \exp\left(\psi(\Omega_t)\right)\right) \tag{12}$$

where, $\psi_t = \boldsymbol{\alpha}_i^w \vec{w}_{g(j),l,t} + \alpha_i^{p,app} p_{g(j),l,t} + \eta_{g(j),l,t}$. Further, as in Hendel and Nevo (2006), note we can interpret $\psi(\Omega)$ as a price index. Next, similar to handsets, to reduce the dimension of the state space we assume that

$$F_{i,g(j)}(\psi_{i,t+1}(\Omega_{t+1})|\Omega_t)) = F_{i,g(j)}(\psi(\Omega_{t+1})|\psi(\Omega_t)), \tag{13}$$

where $F()$ is the (exogeneous) transition process of the state variables.

As before, we assume that with regards to distribution of future states, the consumer only use the lower dimensional price index ψ_t and not the higher dimensional Ω_t. This can be interpreted as a statement about how the actual process Ω_t evolves or as a behavioral statement about how consumers form expectations. The key is that consumers only use the current value of ψ_t to predict the future distribution of ψ_{t+1} instead of the whole information set Ω_t and make download decisions. This reduces the dimension of the state space to a single dimension and makes the computation of W numerically feasible. Specifically, it allows us to rewrite equation (12) as follows

$$W(\psi(\Omega_t)) = \ln(\exp\left(\beta\,\mathbb{E}(W(\psi(\Omega_{t+1}))|\Omega_t)\right) + \exp\left(\psi_t(\Omega_t)\right)) \tag{14}$$

As we did for handsets, we use a AR(1) specification for $\mathbb{E}[\psi_{i,t+1}|\psi_{i,t}]$ as follows:

$$\psi_{i,t+1} = \phi_{i1}^{app,g(j)} + \phi_{i2}^{app,g(j)} \psi_{i,t} + \epsilon_{t+1} \tag{15}$$

where, ϵ_{t+1} is identically and independently distributed with mean 0. ϕ_{i1} and ϕ_{i2} are the incidental parameters.

Using the model we can write an expression for the probability of consumer i to downloaded app l in period t, which is given by

$$s_{ijt}^{app} = \frac{\exp\left(\psi_{i,l,g(j),t}\right)}{\exp\left(\psi_{i,l,g(j),t}\right) + \exp\left(\beta\,\mathbb{E}(W(\psi_{i,l,g(j),t+1}))|\psi_{i,l,g(j),t})\right))} \tag{16}$$

The aggregate share of downloads of app l on platform $g(j)$ at time t can computed by integrating over the distribution of consumers

$$s_{jt}^{app} = \int s_{ijt}^{app} d\mathcal{P}^*$$

where $d\mathcal{P}^*$ denotes the population distribution of consumer preferences. To estimate the parameters of the model, we will map this probability to the observed share of app downloads in the data.

Using this model we can also derive the utility a consumer gets from having access to a certain app store. The value is the sum of the utility from apps currently on the platform and discounted utility of apps that will be available on the app store in the future. Formally, the utility from access to the app store $\Gamma(\cdot)$ is given by

$$\Gamma_{g(j),t}(\alpha_i^p, \boldsymbol{\alpha}_i^w) = \underbrace{[\sum_{l \in \cup_{\tau=0}^{\tau=t} L_{g(j),\tau}} W_{i,g(j),l}(\Omega_{i,t})]}_{\text{Current app utility} \equiv \Delta_{i,g(j),t}^c} + \underbrace{E[\sum_{\tau=1}^{\tau=\infty} (\beta)^\tau (\sum_{l \in L_{g(j),t+\tau}} \mathbf{E}[W_{i,g(j),l}(\Omega_{i,t+\tau})])|\Omega_{i,t}]}_{\text{Expected future app utility} \equiv \Delta_{i,g(j),t}^f} \quad (17)$$

where, $L_{g(j),\tau}$ denotes the set of apps launched in period τ. Similarly, $\Delta_{i,g(j),t}^f$, denotes the expected values of apps that are expected to be launched in future given the current information set $\Omega_{i,t}$. $\Delta_{i,g(j),t}^c$ captures the current app utility, attributed by the apps currently present on the platform.

4.3. Developers' Porting Decisions

We now consider the developers' decision to offer apps on the two platforms. Specifically, we model the porting costs that affect developers ability to offer an app across both platforms.

Consider an app l operational on platform p at time t. The expected profit the app developer expects to make by deciding to not port to platform p' at time t is given by

$$\Pi_{l,t}(d_t \equiv \{p\} \rightarrow \{p\}; \theta) = \mathbb{E}\left[\sum_{\tau > t} (\beta^{\tau - t} D_{l,\tau}^p)|\Omega_{l,t}\right] \quad (18)$$

where $D_{l,\tau}^p$ denote the number of downloads of app l in time period τ on platform p ; $\Omega_{l,t}$ is the app developer's information set at time t; θ denotes the model parameters; and $d_t \equiv \{p\} \rightarrow \{p\}$ denotes the decision to stay on platform p. The above expression is the net present value of total number of downloads the developer is expecting to receive from app l and platform p. We normalize expected profits per download to 1 and thus expected present value of app profit can be expressed as discounted value over future stream of downloads. Average profit per user (APPU) metrics typically involve additional accounting for revenues from in-app purchases (e.g., virtual items for gameplay) and advertising over the course of using the app. By normalizing our estimates against APPU, all apps are treated equally with respect to their means of monetizing users, whether by upfront price (free versus paid) or in-app. The consequence of our normalization is that we can not recover the true dollar value of porting costs but only costs relative to download numbers.

Now lets consider the expected profit for a developer that elects to port an app l operational on platform p to platform p'. We assume if a developer choose to port, it can do so by incurring some porting cost C_l. Once the app is deployed on the other platform it gets another stream of downloads $\{D^{p'}_{t+k+1}, D^{p'}_{t+k+2}, \dots\}$ from platform p', where k refers to the time required to port the app to the other platform. Then, the expected normalized profits from app l, if developer chooses to port (i.e., $d_t \equiv \{p\} \to \{p,p'\}$) to platform p' at time t is given by

$$\begin{aligned}\Pi_{l,t}(d_t \equiv \{p\} \to \{p,p'\};\theta) = \mathbb{E}\Big[\sum_{t<\tau\leq t+k}(\beta^{\tau-t}D^p_{l,\tau})|\Omega_{l,t}\Big] + \\ \mathbb{E}\Big[\sum_{\tau>t+k}(\beta^{\tau-t}(D^{p'}_{l,\tau}+D^p_{l,\tau}))|\Omega_{l,t}\}\}\Big] - C_l(\theta)\end{aligned} \tag{19}$$

Note, the stream of downloads from platform p' starts from period $t+k+1$. We can now use the observed empirical patterns of developer entry to infer the economic cost to port an app to another platform (Apple or Android). Specifically, by observing when a developer chooses to port (or not), we can construct estimates of porting costs. If we observe a ported app on a platform at time $\mathcal{T}$, this implies that the developer would have found it profitable to port $k+1$ periods before (recall k is the development time for porting). That is

$$m_{l,\mathcal{T}-k-1} \triangleq \Pi_{l,\mathcal{T}-k-1}(d_{\mathcal{T}-k-1} \equiv \{p\} \to \{p,p'\};\theta) - \Pi_{l,\mathcal{T}-k-1}(d_{\mathcal{T}-k-1} \equiv \{p\} \to \{p\};\theta) \geq 0 \tag{20}$$

Similarly, for all time periods $\tau < \mathcal{T}-k-1$, developer not electing to port implies the following moment inequalities

$$m_{l,\tau} \triangleq \Pi_{l,\tau}(d_\tau \equiv \{p\} \to \{p,p'\};\theta) - \Pi_{l,\tau}(d_\tau \equiv \{p\} \to \{p\};\theta) \leq 0 \tag{21}$$

5. Estimation

In this section, we discuss estimation of the model(s) presented in the previous section using the data described in Section 3. We estimate the different parts of the model separately.

5.1. Estimation of Demand

We estimate both demand for handsets and apps using aggregate, market-level, data using GMM. We follow the approach used in static models, for example, by Berry et al. (1995) and Nevo (2001), and for dynamic models by Gowrisankaran and Rysman (2012) and Melnikov (2013). The essential idea is to estimate the model parameters by minimizing a GMM objective function that is a sample analog of moments between the (structural) error term, which is a function of parameters, and instruments. As we will discuss below, this allows us to deal with the endogeneity of some variables as well as estimate the parameters that govern the distribution of random coefficients. In this section we discuss some key points of estimation. We provide mode details in Appendix A. We start by explaining our choice of the error term to construct the moment conditions for GMM. Then, we describe our instrumental variables and finally outline the computational steps involved in estimation.

To estimate our model and construct the GMM objective function, we build on the following moment restrictions

$$\mathbb{E}\left[\nu_{j,t}^{hs}(\theta^{hs})|\mathbf{Z}_t^{hs}\right]=0 \qquad \mathbb{E}\left[\nu_{g(j),l,t}^{app}(\theta^{app})|\mathbf{Z}_t^{app}\right]=0 \tag{22}$$

where $\mathbf{Z}_t^{hs}$ and $\mathbf{Z}_t^{app}$ are instrumental variables, and θ^{hs} and θ^{app} include all model parameters of handset and app models, respectively. The econometric error terms ν^{hs} and ν^{app} are defined using the first difference of the unobservable attribute

$$\begin{aligned} \nu_{j,t}^{hs} &= \xi_{j,t}-\xi_{j,t-1}, \quad \forall j \in H_{t-1} \\ \nu_{g(j),l,t}^{app} &= \eta_{g(j),l,t}-\eta_{g(j),l,t-1}, \quad \forall l \in L_{g(j),t-1} \end{aligned} \tag{23}$$

A few remarks are in order. First, the estimation is at the aggregate market level and therefore the individual choice level errors $\epsilon_{i,j,t}^{(hs)}$ and $\epsilon_{i,g(j),l,t}^{(app)}$ that affect consumers' choices of handsets and apps are integrated out. Second, by defining the error term as a first difference, we difference away a headset-specific constant (and an app specific constant in the demand for apps). This significantly simplifies the GMM optimization problem, which can now be carried out by searching only for a handful of model parameters. With the model parameters in hand, we can recover the fixed effects. Third, we have two sets of conditional moment restrictions: one set coming from the handset demand model and another from the apps demand model. In principle, we could stack all the moment conditions together and estimate the parameters by minimizing one GMM objective function. This is computationally quite intense so we estimate the apps demand model first and then use the estimates to estimate the handset demand model. We provide more computational details at the end of this Section and in Appendix A.

Instruments: For instruments $\mathbf{Z}_t^{hs}$ and $\mathbf{Z}_t^{app}$ we use (lagged) values of observable features of the handsets and apps, respectively. Recall that our error term is in first differences. Therefore, we are assuming that the evolution of the error term from period $t-1$ to t, was unexpected when the features were set.

Specifically, the instrument set Z_t^{hs} includes, two- period lagged values of handset price, current- and one- period lagged values of platform utility $\{\Delta_{i,g(j),t}^c, \Delta_{i,g(j),t-1}^c\}$, and current period values of

BLP-style instruments[10] on handset exogenous characteristics – ram, display-size, battery, storage, camera.

From our model, handset prices could be correlated with the shocks in handset demand (ν^{hs}) and hence could be endogeneous. We assume that the two-lagged values of prices do not correlate with ν^{hs} but do correlate with level difference in handset prices, and hence could help identify the price coefficient. With regards to the platform utility, the exclusion of current- and one- period lagged values of platform utility $\{\Delta^c_{i,g(j),t},\ \Delta^c_{i,g(j),t-1}\}$ relies on a timing assumption, namely that software firms cannot immediately respond to shocks in individual handsets' sales or predict them in advance. We test the robustness of our results against the possibility that some app developers might be able to track individual handset sales and respond rapidly in Appendix C and do not find evidence for the same. For instance, we find that using the two-period lagged value of current platform utility $\{\Delta^c_{i,g(j),t-2}\}$ gives almost similar model estimates.

The instrument set Z^{app}_t includes, two- period lagged values of app prices, two- period lagged value of version number and version age, two-period lagged value of app's rating, and current period values of BLP-style instrument on app-age.

Similar to handset prices, app prices might also be correlated with app demand shocks ν^{app}, and be endogeneous. We assume that the two-lagged values of app prices do not correlate with ν^{app} but do correlate with level difference in app prices, and hence could help identify the price coefficient. Also, app's versioning behavior (and thus version age and version #) and app's current period rating may also be correlated with sudden demand shocks. We assume that two-period lagged value of app's versioning related variables and two-period lagged value of app's current period rating does not correlate with demand shocks, but does correlate with level differences in app's version age and version #, and current period rating, respectively.

To test if our instruments are not weak we consider the first stage regressions in level differences of the endogeneous variables onto their respective instrument set. We carry out the weak instrument test as laid out in Cragg and Donald (1993). Since, for both handsets and apps, we have multiple endogenous variables, we compute the Cragg-Donald statistic i.e., the minimum eigenvalue of the matrix analog of F-statistic from the first stage regression. We calculate the critical value for weak

[10] We follow the suggestion of Berry et al. (1995), and for each product j and its exogeneous observable characteristic k we construct the sum of product characteristic k of other products controlled by the same firm f and of those controlled by competing firms within the same market t.

$$\mathcal{H}_j(\vec{x}^k_t) = \Big\{\{\sum_{i\in P_{ft}\setminus\{j\}} x^k_{it},\ \sum_{i\notin P_{ft}} x^k_{it}\}\Big\} \tag{24}$$

where P_{ft} refers to the set of products offered by firm f at time t. For handsets a firm refers to the manufacturer of the handset like Apple, Samsung, etc., and for apps a firm refers to the company that developed the app.

instruments using the procedure laid out in Stock and Yogo (2002). Across all specifications we consider, we are able to reject the hypothesis that our instruments are weak.

Our identification argument assumes that the handset and app specific demand shocks (ν_t^{hs},ν_t^{app}) are serially uncorrelated. To test the validity of this assumption, we use the estimated first differences ($\hat{\nu}_t^{hs}$, $\hat{\nu}_t^{app}$) (for more details see Wooldridge (2010)) from our estimated model and estimate the following regression:

$$\hat{\nu}_t = \kappa\hat{\nu}_{t-1} + error_t \tag{25}$$

Estimating the above regression, we are unable to reject the null that $\kappa = 0$ for both handset and app models.

Computation: Now, we will briefly outline the steps we take for estimating our model parameters. To be able to estimate our model using moment conditions (22), we first need to compute demand shocks ν^{hs} and ν^{app} as a function of the data and model parameters. Broadly, we will follow a similar procedure as used in Berry et al. (1995), with some deviations. The key insight on which our estimation is based is that the unobservable product characteristics for handset and apps only appear in the expression of mean lifetime utilities of handset ($\delta_{j,t}$) and apps ($\psi_{g(j),l,t}$), respectively. Thus, we will first estimate the mean lifetime utilities (δ and ψ) from observed data as a function of model parameters. For this we will use the analytical expressions for purchase probabilities using our model. As described in Section 4, the probability of purchase of particular handset in given period (see equation 8) and the probability of download of an app in a given period (see equation 36) can be written down in terms of model parameters

$$s_{ijt}^{hs} = \frac{\exp(\delta_{i,j,t})}{\exp(\zeta_{i,j,t}) + \exp(\beta\,\mathbb{E}(V(\zeta_{i,j,t+1})|\zeta_{i,j,t})))}$$

$$s_{i,l,g(j),t}^{app} = \frac{\exp(\psi_{i,l,g(j),t})}{\exp(\psi_{i,l,g(j),t}) + \exp(\beta\,\mathbb{E}(W(\psi_{i,l,g(j),t+1}))|\psi_{i,l,g(j),t})))}$$

We can map these expressions to the observed shares and solve the resultant implicit system of equations to solve for the mean lifetime utilities terms δ and ψ

$$s_{\cdot t}^{hs}(\cdot) = S_{\cdot t}^{hs} \qquad\qquad s_{\cdot t}^{app}(\cdot) = S_{\cdot t}^{app}$$

where $S_{\cdot t}^{hs}$ and $S_{\cdot t}^{app}$ are observed aggregate market shares of particular handsets and apps.

In our case, since we cannot analytically integrate over the customer distribution, we numerically compute left hand side of the above equations i.e., the aggregate market shares $s_{\cdot t}$ (see Appendix for details). To model consumer heterogeneity, we assume that price sensitivities of consumers are parameterized as follows: $\alpha_i^{p,l} = \alpha^{p,l} + \sigma^l\nu_i$, for $l \in \{hs, app\}$ and ν_i is drawn from standard normal distributions. To compute the market shares, we also need access to value functions $V(\delta)$ and $W(\psi)$. To compute them we first use an ordinary least square regression to estimate the incidental parameters (ϕ^{hs} and ϕ^{app}) using the currently guessed values of mean utilities δ and ψ. We then compute the value functions $V(\delta)$ and $W(\psi)$ using the imputed values of incidental parameters using a fine grid approximation over δ and ψ. We now carry out a contraction mapping similar to BLP to impute the values of mean utilities from the observed data

$$\delta_{\cdot t}^{h+1} = \delta_{\cdot t}^{h} + \lambda^{hs}(s_{\cdot t}^{hs}(\delta_{\cdot t}^{h}, \phi^{hs}, \sigma^{hs}) - S_{\cdot t}^{hs})$$

$$\psi_{\cdot t}^{h+1} = \psi_{\cdot t}^{h} + \lambda^{app}(s_{\cdot t}^{app}(\psi_{\cdot t}^{h}, \phi^{app}, \sigma^{hs}, \sigma^{app}) - S_{\cdot t}^{app})$$

Note, predicted market shares for each particular app also depends on the heterogeneity parameters of handsets (σ^{hs}). We take a two step estimation procedure where σ_{hs} is estimated first from the handset model. Adding tuning parameters λ^{hs} and λ^{app} allows for faster convergence of contraction mapping. We carry out the above iterations for each model until $|\delta_{\cdot t}^{h+1} - \delta_{\cdot t}^{h}|$ and $|\psi_{\cdot t}^{h+1} - \psi_{\cdot t}^{h}|$ are less than the error tolerance threshold. After the convergence on mean utility the incidental parameters are updated using a ordinary least square regression on the updated mean utility. The contraction mapping iterations are again run to get updated mean utility. We stop once we get convergence on the incidental parameters. With the mean utility in hand, we will use the expressions in equation to compute the unobservable product characteristics as a function of model parameters

$$\xi_{jt} = \hat{\delta}_{j,t} - \vec{x}_{jt}\theta^{hs}$$

$$\eta_{jt} = \hat{\psi}_{g(j),l,t} - \vec{x}_{g(j),l,t}\theta^{app}$$

Our econometeric error terms can then simply by constructed by taking the first difference of ξ and η. Since we know all time invariant characteristics get cancelled out after taking the first difference, we only include the time varying characteristics in $\vec{x}_{j,t}^{hs}$ and $\vec{x}_{g(j),l,t}^{app}$. Finally, the GMM objective function for the handset and app models can be constructed by interacting our econometeric error terms with above specified instruments for each model respectively

$$\arg\min_{\theta^{hs}}(\nu_{j,t}^{hs}(\theta^{hs})^T \mathbf{Z}_t^{hs})\mathbf{W}^{hs}(\nu_{j,t}^{hs}(\theta^{hs})^T \mathbf{Z}_t^{hs})^T$$

$$\arg\min_{\theta^{app}}(\nu_{g(j),l,t}^{app}(\theta^{app})^T \mathbf{Z}_t^{app})\mathbf{W}^{app}(\nu_{g(j),l,t}^{app}(\theta^{app})^T \mathbf{Z}_t^{app})^T$$

where $\mathbf{W}^{hs}$ and $\mathbf{W}^{app}$ are the weighting matrices. We use $\left(\mathbf{Z}^{hs'}\mathbf{Z}^{hs}\right)^{-1}$ and $\left(\mathbf{Z}^{app'}\mathbf{Z}^{app}\right)^{-1}$ as our weighting matrices respectively. The model parameters for both the handset and app models can be estimated by search over parameters that minimize the GMM objective functions. To estimate the model parameters pertinent to time-invariant characteristics we use a second ordinary least square regression.

5.2. Estimation of Developers' Porting Costs

We carry out the estimation of porting cost parameters using methods developed in Pakes et al. (2015). The method is similar to generalized methods of moments (Hansen (1982)) method with a slightly different objective function. We first construct sample analogues of the moment conditions $m_{l,t}$, specified in Section 4 as a function of model parameters. To do so, we use $h(\cdot)$ a set of positive exogeneous functions to aggregate moments to construct population moments for each genre g and platform p as follows

$$\hat{m}_g^p = \frac{1}{n} \sum_{l \in \mathcal{S}_{g,p}} \sum_t h(\Omega_{t,l,p}) \otimes m_{l,t} \tag{26}$$

where $\mathcal{S}_{g,p}$ is the set of apps[11] on platform p, with genre g and $\otimes$ represents the Kronecker product.

To compute the values of moments $(m_{l,t})$, we need to estimate the present value of app profits i.e., $\mathbb{E}[D^p_{l,t'}|\Omega_{l,t}, d_t]$ for $t' > t$. To simulate the demand for apps in future periods, we use the estimated demand processes F and G. App developers know that their launch decisions only affect the levels $\{\delta_{i,j,t}\}$ and not the evolution processes F and G. This allows us to simulate the downloads for each app in the future independent of other app developers' decisions. For our estimation, we assume app developers take on an average 1[12] month[13] to port an app with an already developed back-end infrastructure. With regards to porting cost, we assume the specification for porting cost $C_l(\theta) = \theta^g_p$, where $\theta \equiv \{\theta^g_p\}$, and g is the genre of the title l. Formally, the set estimator $\hat{\Theta}$ that consistently identifies the set Θ_0 is given as follows

$$\hat{\Theta} = \arg\min_\theta ||\hat{\Sigma}_{ii}^{-\frac{1}{2}} \min(0, \hat{m}(\theta))|| \tag{27}$$

where, $\hat{\Sigma} = \mathbb{V}(\hat{m}(\Theta_0))$, i.e., the variance-covariance matrix of moment conditions. The key idea behind the above objective function is that it penalizes all the violations in the moment inequalities.

[11] For our analysis, we only include apps which demonstrate non-exclusivity during their lifetime (i.e., up till Jan 2017). Apps choosing to be exclusive do so for reasons that are not entirely pertinent to economic constraints on porting cost. Since such constraints are not observable to us as researchers, and thus we limit our analysis to the subset of apps which demonstrate non-exclusivity.

[12] We run our analysis for values of $k = 1, 3,$ and 5 months, and find slightly different cost estimates however similar counterfactual implications.

[13] According to many business reports (see for instance Existek (2020)) the average to time to develop the front-end of an app is around $\sim$ 6 weeks.

All we know is that the moments are positive at the true parameters. So the moment inequality estimator penalizes values of θ that lead to negative values of moment and tries to minimize the number of violations. Note that unlike methods that utilize moment equalities, estimating model parameters using inequalities can lead to set estimates, rather than a point estimate. Thus, if multiple values of θ satisfy the moment inequalities (i.e., minimize the equation 27 to 0), all are accepted, however if none satisfies, then θ that minimizes absolute value of deviation in inequalities is admitted. We normalize the moments by the square root of variance of the moments. To compute the variance, we carry out the estimation in two steps. In the first step, we estimate Θ_0 using an identity matrix as the normalizing factor. Using the estimated set $\hat{\Theta}_0$, we then compute $\hat{\Sigma}$. To compute $\hat{\Sigma}$, we use centroid of the identified set $\hat{\Theta}_0$. Next, we re-estimate Θ, using the estimated $\hat{\Sigma}$ as the normalizing factor.

With regards to inference, literature has proposed multiple characterization and methods for defining, and estimating confidence intervals. For our analysis, we define confidence interval as the set that includes each element of the identified set with fixed probability (see Imbens and Manski (2004) and Romano and Shaikh (2010))[14]. We detail the procedure to compute confidence interval in Appendix A.2.

Finally, as a robustness test for moment inequalities, we run a misspecification test as described in Ho and Pakes (2014), to test if the moment inequalities are correctly specified (see Appendix F for details). We were unable to reject the null that our model is correctly specified.

6. Results

Demand Estimates. We first present the estimation results for handset and app demand models. In Table 6, we present results from three different specifications of the demand for handsets. In column (I), we present the results for a model where $\Gamma(\cdot)$ is approximated by the total number of apps (i.e., we assume that each app contributes equally to platform utility). To estimate the model, we use the set of instruments described in Section 5. However, instead of lagged values of platform quality, we use current- and one- period lagged value of platform size (i.e., the number of apps) as an IV. We find that platform size has statistically significant and positive effect on handset demand.

In column (II), we present the results for a static model, where we assume that in each period, consumers choose to either buy one of the available handsets or exit the market without purchasing. They cannot choose to wait in anticipation of a better option in the future. We present more details

[14] Another way literature has characterized confidence interval is, as the set that includes the identified set with a fixed probability (see Chernozhukov et al. (2007)). However, we believe the approach of Imbens and Manski (2004) is more in line with the traditional definition of confidence interval in that they should cover the true value of the parameter with fixed probability.

of the static model of handset demand in Appendix B. In the static model the effect of price on handset demand is not statistically significant and the estimated coefficient is smaller, compared to the estimates in columns I and III, which account for dynamics in consumer behavior. This reinforces the point we made in Sections 3 and 4 that ignoring consumer dynamics could lead to underestimation of demand elasticities.

Finally, in column (III), we report results for a model where we allow for both consumer dynamics and heterogeneity across apps in the impact on platform utility. We estimate the complete model as described in Section 4. We find that platform utility $\Gamma(\cdot)$ has a positive and significant impact on handset demand.

To better interpret the model estimates we compute elasticities with respect to each handset characteristic. To compute these elasticities we change the respective characteristic by 1%, and then use the model estimates to compute the corresponding change in handset demand. We assume consumers expect the specific characteristic to align back next period, and hence the artificial intervention does not affect consumers expectation over future states. This procedure translates to simulating a temporary change in handset characteristic. Another way to interpret the model estimates would be to simulate a permanent change in handset characteristic wherein the specific characteristic does not align back in the next period. In general, we find the permanent elasticity numbers to be a little lower than temporary ones. For the sake of brevity we will only discuss the temporary elasticity numbers here.

We summarize the aggregate elasticity numbers with respect to handset characteristics in Table 7. To compute the aggregate elasticities, we use the estimates from column (III) and average across handsets. For instance, we find that the demand for a particular handset increases by 1.90% with a 1% increase in display size. With respect to RAM, we do not find a significant impact on handset demand.

To compute the price elasticity of handset demand we averaged across all handsets and find that a 1% price increase leads to a decline in sales of the particular handset by 2.64%. In Table 7, we report price elasticities for a few handsets. For instance, consider the own-price elasticity of the iPhone 5, which is among the top purchased handsets. We estimate a sales will fall by 2.65% following a 1% price increase. Similarly, for Samsung Galaxy S4, we find that its sales will drop by 2.60%, with a price increase of 1%. We do not find much heterogeneity in price sensitivities across the consumer population, as indicated by a small but statistically significant σ_{hs} parameter.

In Table 8, we report estimates of app demand parameters, and in Table 9 we present the aggregate elasticities with respect to app characteristics implied by these results. We find a positive and significant effect of app age on app demand i.e., consumers drive higher lifetime utility from apps that are mature in age. This could be driven by the fact that as apps mature, they undergo

Table 6. Estimated Parameters (Handsets)

	I	II	III
log(prices)	-2.082***	-0.093	-2.705***
	(0.453)	(0.266)	(0.722)
ram	-0.575***	-0.093	0.010
	(0.134)	(0.079)	(0.124)
display-size	0.633***	0.401***	0.412***
	(0.149)	(0.088)	(0.138)
battery	0.001***	0.0004***	0.001***
	(0.000)	(0.000)	(0.000)
log(storage)	0.792***	-0.021	0.739***
	(0.054)	(0.032)	(0.050)
camera	0.232***	0.073***	0.233***
	(0.018)	(0.011)	(0.016)
log(#apps)	1.595***	-	
	(0.349)		
Γ	-	0.124***	0.257***
		(0.017)	(0.031)
Handset FE	✓	✓	✓
$\sigma^{p,hs}$	0.003	0.090	0.008***
	(0.881)	0.238	(0.000)
N	895	895	895
GMM Obj	4.76e-4	1.171e-05	1.41e-4

Standard errors in parentheses. *p<0.1; **p<0.05; ***p<0.01. For specification (I) we estimate a model where $\Gamma(\cdot)$ is approximated by the total number of apps (i.e., we assume that each app contributes equally to platform utility). For specification (II) we estimate a model that does not account for dynamics in consumer buying behavior. For specification (III) we estimate our full model, that allows for heterogeneity in app externality and consumer dynamics. The parameters for time-invariant characteristics are estimating by projecting the estimated handset fixed effects onto the handset time-invariant characteristics using ordinary least square regression.

Table 7. Handset Elasticities

log(price)	-2.64%
display size	1.90%
battery	2.26%
log(storage)	0.73%
camera	2.08%
Price Elasticity Across Handsets	
iPhone 5	-2.65%
Samsung Galaxy S4	-2.60%
LG G2	-2.63%

Note. The numbers report the percentage change in handset sales following a temporary change in respective handset characteristic by 1%. Temporary change implies that customers belief over future evolution remains unchanged following a price reduction.

a series of experiments and development rounds by its developers, which improve its quality and make it more competitive in the marketplace. Further, we also find a negative and significant effect of version age on app demand i.e., as the current version of an app gets old (i.e., the app is not updated in time), it becomes less attractive to consumers. This indicates that customers prefer apps by developers who regularly update their apps and engage in quality versioning. However, we also find evidence that controlling for overall age and versioning age, apps having a relatively low frequency of versioning are preferred. We find negative and significant effect of version-# on app's demand. A very high number of versioning rounds could indicate lower quality levels and error proneness on the developer side. Hence, a preferred app developer is the one that engages optimally in quality versioning with regular updates to cater to evolving consumer needs. Finally, we estimate that paid Apple and Google Play apps demonstrate price elasticities of -1.51% and -2.03% respectively. Our results are consistent with many business reports, that suggest Android consumers are much more price-sensitive than the Apple consumers.

In Table 8b, we report how time-invariant app characteristics affect app demand (and indirectly handset demand). We regress the estimated app fixed effects on the various time invariant app characteristics e.g, if app has in app purchases (*in-app-purchases*), if the app is paid (*paid*), and genre of the app. We find that social networking apps tend to attract highest demand. Also paid apps tends to attract much lower demand (and thus, exert lower network externalities) than free apps. Thus, platforms operators, should be vary of the platform fees charged by them from the

Table 8. Estimated Parameters (Apps)

(a)

	(I)
price	-0.540***
	(0.076)
version age	-0.007***
	(0.0014)
version #	-0.083***
	(0.019)
ratings	6.355***
	(0.822)
age	0.002***
	(0.000)
App FE	✓
$\sigma^{p,app}$	0.145***
	(0.005)
N	1,801,815
GMM Obj	6.52e-11

Standard errors in parentheses.
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

(b)

	App Fixed Effects
	(1)
in-app-purchases	-0.129***
	(0.004)
paid	-2.967***
	(0.004)
$log(firmsize)$	0.232***
	(0.001)
Entertainment	0.757***
	(0.016)
Games	-1.848***
	(0.015)
Lifestyle	0.215***
	(0.017)
Music	0.692***
	(0.017)
Others	0.053***
	(0.015)
Photo and Video	0.586***
	(0.016)
Social Networking	1.781***
	(0.016)
Utilities	0.156***
	(0.016)
ios	-0.667***
	(0.004)
Intercept	-17.72***
	(0.016)
Observations	1801814.0
R2	0.381
Note:	*p<0.1; **p<0.05; ***p<0.01

The estimation is carried on our entire sample of 13,731 iOS and 6,950 Google Play apps. The parameters for time-invariant characteristics are estimating by projecting the estimated app fixed effects onto the app time-invariant characteristics using ordinary least square regression

apps, as a higher fees could mean highly priced apps, and could reduce demand for their handsets. Finally, apps that make consumers pay for certain functionalities through in app purchases also attract lower demand.

We now turn to the questions of the impact of platform size (or number of apps) on handset demand. For this, we again look at columns I and III in Table 6. Averaging across both platforms, increasing the number of apps by 1% increases the sales of a particular platform on an average by

Table 9. App Elasticities

price	-1.41%
log(firmsize)	0.23%
version age	-0.15%
version #	-0.55%
age	0.79%

Note. The numbers report the percentage change in app downloads following a temporary reduction in respective app characteristic by 1%. Temporary price reduction implies that customers belief over future evolution remains unchanged following a price reduction. We only report numbers for app characteristics that have continuous values.

1.5%. In column (III), we account for heterogeneity in app externalities and estimate the complete model detailed in Section 4. We estimate that 1% increase in platform utilities increases the sales of a particular platform by 1.3%. We also find evidence for significant heterogeneity in externalities apps exert. For instance, we find that the externality exerted by the top app is around 2700 and 5600 times that of the median app on iOS and Google Play platform respectively.

Finally, we look at the impact of top apps on handset demand. Understanding the impact of apps on handset demand could prove crucial to platform operators. [15] In Table 10, we display the apps on Apple and Android platform that have the largest estimated impact on their platform sales. These numbers are computed by removing each app from the available set of apps on the platform. We assume our intervention does not affect consumers belief ($\hat{G}$) over the evolution of handset qualities and hold it constant. We observe that the Social Media apps like Facebook have the highest estimated impact on both the platforms. Removing Facebook from Apple or Android can reduce the demand of respective handsets by -5.9% and -5.6%. We also observe that apps owned by Apple's competitor Google like Youtube (-3.5%) have a significantly high impact on Apple's handset sales.

[15] For instance, many countries have recently considered putting sanctions on certain apps. United States banned WeChat (see `https://www.usatoday.com/story/tech/2020/08/07/what-is-wechat-why-trump-wants-ban-tencent/3319217001/` for United States users and the ban extended to Google and Apple app stores offering the app.

Table 10. Apps with largest estimated impact and value

		(Apple)	Value($)
Apple	Facebook	-5.940	13.04
	Instagram	-5.195	11.40
	Youtube	-3.436	7.54
	Facebook-Messenger	-3.012	6.61
		(Android)	Value ($)
Google Play	Facebook	-5.626	8.58
	Facebook-Messenger	-4.840	7.38
	Instagram	-3.250	4.95
	Snapchat	-1.630	2.48

Note. The table reports the top four apps with highest estimated impact on both Apple and Google Play. The numbers indicate the percentage drop in handset sales following the removal of the app from platform. The reports are estimated by simulating a permanent removal of apps from either platform. The second column shows the estimated impact on the competing platform. The second column report the dollar value as perceived by customers on Apple and Android for one month of app availability.

Our model allows us to factor in the two sided nature of the market and measure the dollar values that consumers attach to free apps (or goods) like Facebook or Instagram. Prior literature has struggled to measure the economic value of free goods or services like Facebook and has mainly relied on customer surveys (for instance Brynjolfsson et al. (2019)) to measure their dollar value. However, our model allows us to infer these values from purchase decisions made by consumers. In Table 10, we report the dollar value for one month access to top apps perceived by both Apple and Android customers.[16] Using our model we find that one month access to Facebook app is worth to be around \$13 (or \$9) to Apple (or Android) users. These numbers are a little lower than \$48 for one month access as reported by Brynjolfsson et al. (2019) using survey data. This is in line with existing literature on survey methods, that has found that consumers to tend to overstate their value for products in surveys (Hausman 2012).

Porting Cost Estimates In Table 11, we show our estimated porting cost to Android and iOS respectively. The estimated costs are normalized to number of downloads. Note, that instead of point estimates, we recover set estimates of developers' porting costs. On an average we find that

[16] One limitation of our analysis is that customers can still access apps through other channels, for example their computers. However, since the majority of traffic, especially to social media websites comes through apps, we can still get very reasonable estimates of their value to consumers. For instance 80% of users access Facebook only through mobile and Instagram was "app only" during the time of our study.

Table 11. Estimated Porting Costs

	Port to Android	Port to Apple
Entertainment	(16875,17175)	6733
	[16375,20375]	[5233,11233]
Games	(18150,18850)	15095
	[16250,23250]	[12095,19595]
Music	(18750,19580)	18805
	[17250,23250]	[13805,23305]
Photo and Video	(31950,34450)	19550
	[27250,36750]	[18550, 23550]
Social Networking	(25350,27750)	28369
	[21250,30750]	[23369,32869]
Utilities	13957	6907
	[13957, 14457]	[5907, 11407]
Others	16874	12230
	[13857, 21375]	[7230, 17230]

Note. We report in round brackets, the identified set of costs to port an app to Android or Apple across the various genres. The estimated costs are normalized to the number of downloads. The 95% confidence intervals are reported in square brackets.

development costs for Android are much higher than for Apple iOS. This is in line with existing literature (e.g. He et al. (2018)), that points out how dealing with multiple prevailing operating system versions of Android, increases development costs for developers. Further, many business reports (for instance see Car (2015)) have also indicated how Android programming requires more written code for same functionality than Apple and can add to development costs. We estimate porting cost to Android to be on an average 33% higher[17]. Our estimates are in the same ranges as reported by many media reports (for instance see Car (2015)). We also find that on an average "Photo and Video" and "Social Networking" apps incur much higher porting costs consistently across both iOS and Android. This is probably due to additional complexity that comes with developing "Photo and Video" and "Social Networking" apps. Both "Photo and Video" and "Social Networking" apps, involve dealing with camera applications. Since, developing *optimized* camera applications, might involve explicitly accounting for various camera hardware available across

[17] We use the center of the identified set for comparison purposes.

Android and Apple handsets, it might add additional cost to developer effort. Further, social media apps have an additional component of handling social networks, which could add to developers cost.

7. Implications

Our empirical results demonstrate substantial heterogeneity in the impact of apps on handset demand. In this section, we discuss (i) how (much) do the preferences of app developers towards platforms affect the platform's attractiveness to consumers, and (ii) could cross-platform development frameworks work as an effective lever for Android to steer developers towards it?

7.1. Effect of Developers' Delayed Entry

Many businesses do not find it profitable to launch apps for both iOS and Android at once, so they start with one platform. For instance, the Android version of Airbnb was launched 14 months after the iOS version. Similarly, Clubhouse is only available on iOS at the time of this writing. This is partly due to resource constraints and partly because businesses often begin with an experimental version to gauge customers' preferences and tend to use only one platform for this initial experimentation.

In our sample, 2103 apps launched on both Apple and Google Play (up to the period two years after the end of the data we use for estimation, i.e., up to Jan 2017). However, the vast majority of app developers launch on Apple first. To understand the economic impact of developers' preference for Apple iOS over the Android platform, we estimate how much more sales Android handsets could have generated (over the period of our study) if apps that launched on Apple first were instead launched on Android at the same time. We then compare the counterfactual demand to the actual realized demand of Android handsets in 2013-2014 to ascertain the losses due to developer's sequential entry.

To simulate the counterfactual demand of Android handsets, we artificially add new apps to the Android platform as they arrive on Apple platform and compute the corresponding demand. One caveat in artificially adding apps to the Android platform is ascertaining the quality levels of new apps to be added. For this purpose, we non-parametrically estimate the Google Play quality (ψ^t_{play}) at any time t, as a function of iOS quality (ψ^t_{ios}), genre and time. Using this we simulate the quality of an app if it was released at earlier time on Android.

We estimate that Android suffers significant losses due to developers preferring Apple for their initial launch. Specifically, we estimate revenue losses from handset sales to be around \$208M in 2013 and \$260M in 2014.

7.2. Role of App Development Frameworks

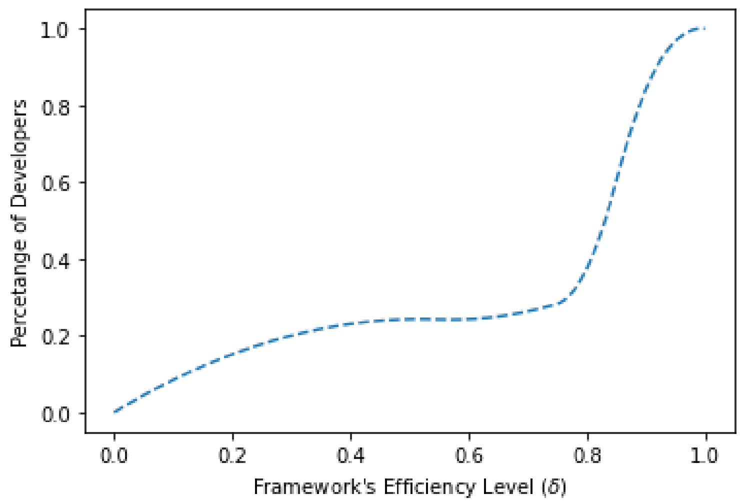
To make the Android platform more attractive to developers, and to counter the effect of developers' preference for iOS, Android operators have started investing in developing cross-platform development frameworks (Heitkötter et al. (2012), El-Kassas et al. (2017)). Cross-platform frameworks enable developers to develop an app once but deploy simultaneously on both Android and Apple. Generally, developers develop *native* apps for one platform and the framework automatically generates apps compatible on other platforms at no additional effort. However, the app generated for the other platform is generally of lower quality (see Mercado et al. (2016)) than if it had been developed natively. Even though many cross-platform frameworks have come up, their impact is not well understood. For instance, it is not well established whether such frameworks can change developers' behavior and to what extent this depends on the efficiency level of the framework. Having a high quality framework might persuade developers to build on Android first and thereby give Android early access to apps whose entry to Android might otherwise be delayed. However, it will also give developers preferring Android early access to the Apple platform. Hence cross-platform frameworks might lead to less differentiated platforms and more intense competition between handsets. This might have adverse consequences for Android demand. By modeling the interconnected nature of customer and developer preferences demand, we can quantify the role these frameworks could play in alleviating loss of demand for Android.

We consider a scenario in which a cross-platform framework allows developers who develop native apps on Android to port their app to iOS but at a slightly lower quality (we let δ denote the level of quality dilution). The Android platform benefits, as it now does not have to suffer the consequences of delayed entry.

To estimate the impact of a cross-platform framework, we allow developers that chose to launch first on Apple to either develop a native app for Android and δ-diluted quality app on iOS, or to continue developing for iOS first and launch on Android at a later date (as observed in our data). We simulate the developers' decisions by comparing their expected profits in the two cases. Developers that launched their app on Android first also get access to the apple platform at the same time as Android although at a lower quality level until they develop a native iOS app.[18] In Figure (12), we highlight developers' decisions as a function of the quality of the cross-platform development framework δ. If the framework can generate a native-quality app on iOS (i.e., $\delta = 1$), all Apple-First developers switch to developing for Android first and use the framework for developing a native app for iOS. However, we find even with a high-quality framework, most developers prefer to develop for iOS first. For e.g., only $\sim 37\%$ of the Apple-First developers choose to switch at

[18] In our simulation, we replace the suboptimal quality app on Apple with a native quality, as and when observed in data.

Figure 12. Estimated Impact on Developers Decision.



Note. The plot shows the percentage of apps originally launched on apple first that switched to using the cross-platform framework as a function of cross-platform framework's efficiency level. The median value of cost's identified set (for each genre) was used to simulate developers' decisions.

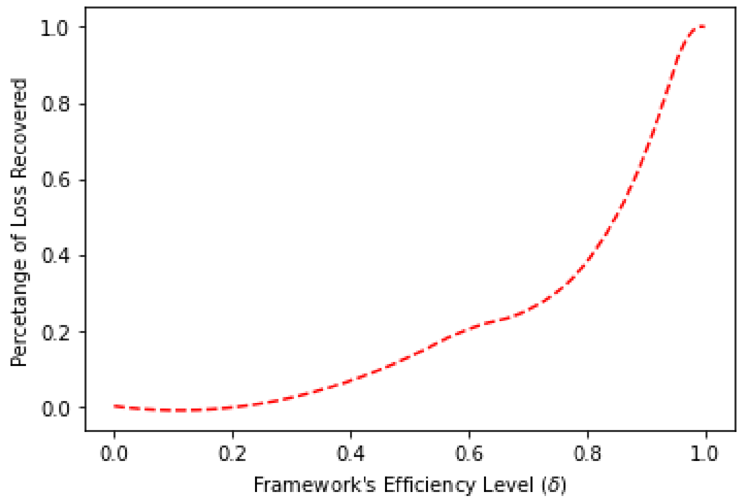
$\delta = 0.8$. This suggests that for the android platform to attract developers, a near-optimal quality framework is required. Further, this also explains, why even with the prevalence of many such development suites, developers still prefer to develop native apps for iOS first.

Next, we quantify how changes in developer behavior affect consumer handset demand. Figure (13) quantifies how the quality of the cross-platform framework affects the percentage of lost sales that can be recovered by Android. As discussed earlier, since most developers tend not to switch their behavior, unless the framework is near-optimal quality, cross-platform frameworks are unable to alleviate much of the loss in sales. For instance, even at 80% efficient framework, only a little under 40% of the lost sales are recovered. Hence, investing in cross-platform frameworks might not deliver anticipated results unless the framework can help create near-native-like apps.

8. Concluding Remarks

In this paper, we investigate the implications of indirect network effects and delayed developer entry on mobile markets. We develop a dynamic model of demand for handsets and apps and use a unique panel dataset of handset sales and app downloads to estimate the model. We find significant heterogeneity in externalities exerted by apps on handset sales. The model helps us quantify the very significant (negative) impact of delayed developer entry on Android. Finally, we find that cross-platform frameworks offered by Android are unlikely to steer developers towards developing on Android first unless they can generate near-native quality iOS apps.

Figure 13. Estimated Sales Recovered.



Note. The percentage of lost sales recovered as a function of cross-platform framework's efficiency levels.

Our findings have many implications for platform operators and policymakers. For platform operators, our findings show how important it is for platforms to detect trending apps that are exclusively available on competing platforms and to court those developers so as to minimize entry delays. This could explain why Microsoft decided to insource[19] development of the Facebook app for its mobile platform back in the day.

Our modeling approach and specific findings are potentially also relevant to several issues that surround competition between app platforms. For example, our results demonstrate that there is significant heterogeneity in the externalities various apps exert on handset sales and therefore in their value to the platform ecosystem. Furthermore, our estimates of porting cost highlight the contrasting difference in porting costs across the two platforms.

Finally, we also want to point limitations of our analysis. In our model, we do not allow for substitution across apps and compromise it to account for consumer dynamics. However, we demonstrate that in our case, a dynamic model demand could lead to much better out-of-sample predictive performance than a model that only accounts for substitution. Incorporating both substitution and dynamics requires keeping track of all possible consumption sets over a large number of products, and hence is not computationally tractable. We believe, methods to tractably solve such combinatorially hard problems could be an interesting direction to work on in the future.

[19] See `https://blogs.windows.com/devices/2013/03/22/facebook-apps-on-your-nokia-lumia/`

References

Aguirregabiria, V. and Nevo, A. (2013). Recent developments in empirical io: Dynamic demand and dynamic games. Advances in Economics and Econometrics, 3:53–122.

Allon, G., Askalidis, G., Berry, R., Immorlica, N., Moon, K., and Singh, A. (2019). When to be agile: Ratings and version updates in mobile apps. Available at SSRN 3476286.

Armstrong, M. (2006). Competition in two-sided markets. The RAND Journal of Economics, 37(3):668–691.

Berry, S., Levinsohn, J., and Pakes, A. (1995). Automobile prices in market equilibrium. Econometrica: Journal of the Econometric Society, pages 841–890.

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

Brynjolfsson, E., Collis, A., and Eggers, F. (2019). Using massive online choice experiments to measure changes in well-being. Proceedings of the National Academy of Sciences, 116(15):7250–7255.

Brynjolfsson, E., Eggers, F., and Gannamaneni, A. (2018). Measuring welfare with massive online choice experiments: A brief introduction. In AEA Papers and Proceedings, volume 108, pages 473–76.

Car, T. (2015 (accessed October, 2015)). Android Development Is 30% More Expensive Than iOS. And We — Infinum.

Chernozhukov, V., Hong, H., and Tamer, E. (2007). Estimation and confidence regions for parameter sets in econometric models 1. Econometrica, 75(5):1243–1284.

Choi, J. P. (1996). Do converters facilitate the transition to a new incompatible technology? a dynamic analysis of converters. International Journal of Industrial Organization, 14(6):825–835.

Church, J., Gandal, N., and Krause, D. (2008). Indirect network effects and adoption externalities. Review of Network Economics, 7(3).

Clements, M. T. (2004). Direct and indirect network effects: are they equivalent? International Journal of Industrial Organization, 22(5):633–645.

Clements, M. T. and Ohashi, H. (2005). Indirect network effects and the product cycle: video games in the us, 1994–2002. The Journal of Industrial Economics, 53(4):515–542.

Conlon, C. T. (2012). A dynamic model of prices and margins in the lcd tv industry.

Corts, K. S. and Lederman, M. (2009). Software exclusivity and the scope of indirect network effects in the us home video game market. International Journal of Industrial Organization, 27(2):121–136.

Cragg, J. G. and Donald, S. G. (1993). Testing identifiability and specification in instrumental variable models. Econometric Theory, pages 222–240.

David, P. A. (1985). Clio and the economics of qwerty. The American Economic Review, 75(2):332–337.

Economides, N. and Salop, S. C. (1992). Competition and integration among complements, and network market structure. Journal of Industrial Economics, 40(1):105–23.

El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., and Wahba, A. M. (2017). Taxonomy of cross-platform mobile applications development approaches. Ain Shams Engineering Journal, 8(2):163–190.

Existek (2020). App development timeline: How long does it take?: Existek blog. Existek Blog.

Farrell, J. and Saloner, G. (1985). Standardization, compatibility, and innovation. The RAND Journal of Economics, 16(1):70–83.

Gabszewicz, J. J. and Wauthy, X. (2004). Two-sided markets and price competition with multi-homing. Available at SSRN 975897.

Gandal, N., Kende, M., and Rob, R. (2000). The dynamics of technological adoption in hardware/software systems: The case of compact disc players. The RAND Journal of Economics, pages 43–61.

Garg, R. and Telang, R. (2013). Inferring app demand from publicly available data. MIS Quarterly, pages 1253–1264.

Ghose, A., Goldfarb, A., and Han, S. P. (2012). How is the mobile Internet different? search costs and local activities. Info. Systems Res., 24(3):613–631.

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

Gowrisankaran, G. and Rysman, M. (2012). Dynamics of consumer demand for new durable goods. J. Political Econom., 120(6):1173–1219.

Hagiu, A. (2006). Pricing and commitment by two-sided platforms. The RAND Journal of Economics, 37(3):720–737.

Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. Econometrica: Journal of the Econometric Society, pages 1029–1054.

Hausman, J. (2012). Contingent valuation: from dubious to hopeless. Journal of Economic Perspectives, 26(4):43–56.

He, D., Li, L., Wang, L., Zheng, H., Li, G., and Xue, J. (2018). Understanding and detecting evolution-induced compatibility issues in android apps. In 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE), pages 167–177. IEEE.

Heitkötter, H., Hanschke, S., and Majchrzak, T. A. (2012). Evaluating cross-platform development approaches for mobile applications. In International Conference on Web Information Systems and Technologies, pages 120–138. Springer.

Hendel, I. and Nevo, A. (2006). Measuring the implications of sales and consumer inventory behavior. Econometrica, 74(6):1637–1673.

Ho, K. and Pakes, A. (2014). Hospital choices, hospital prices, and financial incentives to physicians. American Economic Review, 104(12):3841–84.

Imbens, G. W. and Manski, C. F. (2004). Confidence intervals for partially identified parameters. Econometrica, 72(6):1845–1857.

Lee, R. S. (2013). Vertical integration and exclusivity in platform and two-sided markets. American Economic Review, 103(7):2960–3000.

Li, X., Bresnahan, T., and Yin, P.-L. (2016). Paying incumbents and customers to enter an industry: Buying downloads. Available at SSRN 2834564.

Melnikov, O. (2013). Demand for differentiated durable products: The case of the us computer printer market. Economic Inquiry, 51(2):1277–1298.

Mercado, I. T., Munaiah, N., and Meneely, A. (2016). The impact of cross-platform development approaches for mobile applications from the user's perspective. In Proceedings of the International Workshop on App Market Analytics, pages 43–49. ACM.

Nair, H., Chintagunta, P., and Dubé, J.-P. (2004). Empirical analysis of indirect network effects in the market for personal digital assistants. Quantitative Marketing and Economics, 2(1):23–58.

Nevo, A. (2001). Measuring market power in the ready-to-eat cereal industry. Econometrica, 69(2):307–342.

Ohashi, H. (2003). The role of network effects in the us vcr market, 1978–1986. Journal of Economics & Management Strategy, 12(4):447–494.

Pakes, A. (2012). Nber econometrics minicourse. National Bureau of Economic Research.

Pakes, A., Porter, J., Ho, K., and Ishii, J. (2015). Moment inequalities and their application. Econometrica, 83(1):315–334.

Park, S. (2004). Quantitative analysis of network externalities in competing technologies: The vcr case. Review of Economics and Statistics, 86(4):937–945.

Parker, G. G. and Van Alstyne, M. W. (2005). Two-sided network effects: A theory of information product design. Management Science, 51(10):1494–1504.

Rochet, J.-C. and Tirole, J. (2003). Platform competition in two-sided markets. Journal of the European Economic Association, 1(4):990–1029.

Romano, J. P. and Shaikh, A. M. (2010). Inference for the identified set in partially identified econometric models. Econometrica, 78(1):169–211.

Roodman, D. (2007). A short note on the theme of too many instruments. Center for Global Development Working Paper, 125.

Rust, J. (1994). Structural estimation of Markov decision processes. In Engle, R. and McFadden, D., editors, Handbook of Econometrics, volume 4, chapter 51, pages 3081–3143. North Holland.

Sen, S., Jin, Y., Guérin, R., and Hosanagar, K. (2010). Modeling the dynamics of network technology adoption and the role of converters. IEEE/ACM Transactions on Networking, 18(6):1793–1805.

Shankar, V. and Bayus, B. L. (2003). Network effects and competition: An empirical analysis of the home video game industry. Strategic Management Journal, 24(4):375–384.

Shapiro, C. and Varian, H. R. (1998). Versioning: The smart way to sell information. Harvard Business Rev., 107(6):107.

Stock, J. H. and Yogo, M. (2002). Testing for weak instruments in linear iv regression. Technical report, National Bureau of Economic Research.

Wang, Q., Li, B., and Singh, P. V. (2018). Copycats vs. original mobile apps: A machine learning copycat-detection method and empirical analysis. Information Systems Research, 29(2):273–291.

Wooldridge, J. M. (2010). Econometric Analysis of Cross Section and Panel Data. MIT press.

Appendices

A. Estimation Details

A.1. Estimation of Demand Model

As described in Section 5, we use generalized method of moments to estimate our demand model for both handsets and apps. We construct the GMM objective function using equations (22) and (23). We use a nested estimation procedure, which simplifies the estimation somewhat. We use a a nested estimation procedure because we need customer heterogeneity parameters of handsets to estimate apps demand parameters. This is because, the distribution of consumers that can download apps is dependent on the type of customers who buy the handset. The steps of our estimation procedure are as follows.

1. **Initialization Phase**: In this stage, we get an initial estimate of customer heterogeneity ($\sigma^{p,hs}$) over handset prices. For initialization purpose, we will not take into account the platform utility Γ. The mean of the platform utility is captured by the handset fixed effects $\bar{\xi}_j$, and the deviations from mean are captured by the unobserved handset utility term $\Delta\xi_{j,t}$. To estimate the handset specific parameters, we minimize the following GMM objective function

$$\arg\min_{\theta^{hs}}(\nu_{j,t}^{hs}(\theta^{hs})^T\mathbf{Z}_t^{hs})\mathbf{W}^{hs}(\nu_{j,t}^{hs}(\theta^{hs})^T\mathbf{Z}_t^{hs})^T$$

where $\nu_{j,t}^{hs}(\theta) = \xi_{j,t} - \xi_{j,t-1}$. We use the instruments as detailed in Section 5, except the lagged values of platform utility (Γ). Next, to estimate $\xi_{j,t}$, we consider the following set of equations. Consider δ_{jt} as the *mean lifetime utilities* of handset for consumers with mean parameter values $\boldsymbol{\alpha}^{p,hs}$. By definition of mean lifetime utilities (see equation (1)) we have

$$\begin{gathered}\delta_{j,t}(\theta) = \boldsymbol{\alpha^x} x_{j,t} + \boldsymbol{\alpha^{p,hs}} \log(p)_{j,t} + \xi_{j,t} \\ \xi_{j,t} - \xi_{j,t-1} = \delta_{j,t}(\theta) - \delta_{j,t-1}(\theta) - \boldsymbol{\alpha^x}(x_{j,t} - x_{j,t-1}) - \\ \alpha^{p,hs}(\log(p)_{j,t} - \log(p)_{j,t-1})\end{gathered}$$

Thus, we have

$$\begin{gathered}\nu_{j,t}^{hs}(\theta) = \delta_{j,t}(\theta) - \delta_{j,t-1}(\theta) - \boldsymbol{\alpha^x}(x_{j,t} - x_{j,t-1}) - \\ \alpha^{p,hs}(\log(p)_{j,t} - \log(p)_{j,t-1})\end{gathered}$$

As one can note, the handset demand shocks are a function of handset mean parameters, handset observed time-varying characteristics and handset *mean lifetime utility* δ_{jt}.

Customer Heterogeneity: To simulate customer heterogeneity, we assume that customer price taste parameters, at the start of our study are distributed as independent normal random variables (i.e., $\boldsymbol{\alpha}_i^p - \boldsymbol{\alpha}^p \sim N(0, \Sigma)$),

$$\Sigma = \begin{bmatrix} \sigma_{hs} & 0 \\ 0 & \sigma_{app} \end{bmatrix} \tag{28}$$

Thus,

$$\boldsymbol{\alpha}_i^{p,hs} = \boldsymbol{\alpha}^{p,hs} + \nu_i^{hs} \sigma_{p,hs} \text{ and } \boldsymbol{\alpha}_i^{p,app} = \boldsymbol{\alpha}^{p,app} + \nu_i^{app} \sigma_{app} \text{[20]} \tag{29}$$

Also, note that $\delta_{i,j,t}$ can also be written as follows

$$\delta_{i,j,t} = \delta_{j,t} + (\boldsymbol{\alpha}_i^{p,hs} - \boldsymbol{\alpha}^{p,hs})(\log(p)_{j,t}) \tag{30}$$

Also as discussed in Section 4, the logit inclusive value $\zeta_{i,j,t}$ and mean lifetime utility for handset, $\delta_{i,j,t}$ are sufficient statistics for determining if a consumer of type i will purchase a handset j at time t or not. The probability of consumer type i, to buy handset j in period t is given by

$$\hat{s}_{i,j,t}(\delta_{i,j,t}, \zeta_{i,j,t}) = \frac{\exp(\zeta_{i,j,t})}{\exp(\zeta_{i,j,t}) + \exp(\beta\, \mathbb{E}(V(\zeta_{i,j,t+1})|\zeta_{i,j,t})))} * \frac{\exp(\delta_{i,j,t})}{\exp(\zeta_{i,j,t})} \tag{31}$$

However, in our data we only observe aggregate shares for each handset. Thus, we numerically approximate the normally distributed customers as a finite distribution of consumer type i so that each type has a corresponding weight w_{it}. This also allows us to update the distribution of customers when they exit after making a purchase by adjusting the weights.

$$\begin{aligned}
\hat{s}_{j,t}(\delta_{i,j,t}, \zeta_{i,j,t}) &= \sum_i w_{it}^{hs} \hat{s}_{i,j,t}(\delta_{i,j,t}, \zeta_{i,j,t}) \\
\sum_i w_{i0}^{hs} &= 1 \\
w_{it+1}^{hs} &= w_{it}^{hs}(1 - \sum_{j \in H_t} s_{ijt})
\end{aligned}$$

For the initial consumer distribution, the weights sum upto 1. Ideally one can use initially weights to be $\frac{1}{n_I}$, where I is the number of individuals to be simulated. However, we use the points (i.e., (w_i, ν_i)) generated through gauss hermite quadrature rule to get better approximation of the consumer distribution and simulate customer type i. Note, that to estimate market shares we also need access to the term $\mathbb{E}(V(\zeta_{i,j,t+1})|\zeta_{i,j,t}))$. To estimate the expected value of the value function, we will first estimate the incidental parameters ϕ_{i1} and ϕ_{i2} using a first stage regression based on current guess of ζ

$$\zeta_{i,t+1} = \phi_{i1} + \phi_{i2}\zeta_{i,t} + \mu_{it+1} \tag{32}$$

[20] Since few paid apps were launched before our period of study, $\alpha^{\mathbf{p,app}}$ and σ_{app} could potentially vary across them. However, for numerical tractability we impose $\alpha_{\mathbf{app}}^{\mathbf{l}} \approx \alpha_{\mathbf{app}}$, and $\sigma_{app}^{l} \approx \sigma_{app}$ $\forall l$.

We then compute the value functions $V(\zeta)$ using the imputed values of incidental parameters We use a fine grid approximation over ζ and use value iteration to estimate the value function V. Next, similar to Berry et al. (1995), we use contraction mapping to estimate $\vec{\delta}$ given $\sigma_{p,hs}$ from the observed market shares $S^{hs}_{\cdot t}$.

$$\delta^{h+1}_{\cdot t} = \delta^{h}_{\cdot t} + \lambda^{hs}(s^{hs}_{\cdot t}(\delta^{h}_{\cdot t}, \phi^{hs}, \sigma^{hs}) - S^{hs}_{\cdot t})$$

Adding tuning parameters λ^{hs} allows for faster convergence of contraction mapping. We carry out the above iterations until $|\delta^{h+1}_{\cdot t} - \delta^{h}_{\cdot t}|$ are less than the error tolerance threshold. After the convergence on mean utility, ζ is updated using the values of δ. Then, the incidental parameters are updated using ordinary least square regression in equation 32. The contraction mapping iterations are again run to get updated mean utility. We stop once we get convergence on the incidental parameters (ϕ_{i1} and ϕ_{i2}). With the mean utility in hand, we will use the expressions in equation below to compute the unobservable product characteristics as a function of model parameters

$$\xi_{jt} = \hat{\delta}_{j,t} - \vec{x}_{jt}\theta^{hs}$$

Our econometeric error terms can then simply by constructed by taking the first difference of ξ. Since we know all time invariant characteristics get cancelled out after taking the first difference, we only include the time varying characteristics in $\vec{x}^{hs}_{j,t}$. Finally, the GMM objective function for the handset and app models can be constructed by interacting our econometeric error terms with above specified instruments for each model respectively

$$\arg\min_{\theta^{hs}}(\nu^{hs}_{j,t}(\theta^{hs})^T \mathbf{Z}^{hs}_t)\mathbf{W}^{hs}(\nu^{hs}_{j,t}(\theta^{hs})^T \mathbf{Z}^{hs}_t)^T \tag{33}$$

where $\mathbf{W}^{hs}$ is the weighting matrix. We use $(\mathbf{Z}^{hs\prime}\mathbf{Z}^{hs})^{-1}$ as our weighting matrix. The model parameters for handset model can then be estimated by search over parameters that minimize the GMM objective functions. To estimate the model parameters pertinent to time-invariant characteristics we use a second ordinary least square regression with handset level fixed effects as the dependent variable and time-invariant characteristics as the independent variables. Finally, we update the σ^{hs} parameter, and re-estimate our model. We continue iterating until we find the parameter σ^{hs} that minimizes the GMM objective criterion.

2. **App Demand**: In this phase once we have estimated customer heterogeneity parameters for handset parameters, we move to estimate model parameters for apps. Similar to the handset side, we use generalized method of moments for estimating model parameters on the app side.

We know $\nu^{app}_{g(j),l,t}(\theta) = \eta^{app}_{g(j),l,t}(\theta) - \eta^{app}_{g(j),l,t-1}(\theta)$. Thus, to estimate $\nu^{app}_{g(j),l,t}(\theta)$, we have

$$\nu^{app}_{g(j),l,t}(\theta) = \psi_{g(j),l,t}(\theta) - \psi_{g(j),l,t-1}(\theta) - \boldsymbol{\alpha}^{\boldsymbol{w}}(w_{g(j),l,t} - w_{g(j),l,t-1}) - \alpha^{p,app}(p_{g(j),l,t} - p_{g(j),l,t-1}) \tag{34}$$

where $\psi_{g(j),l,t}(\theta)$ denotes the *mean lifetime utilities* consumers get from downloading or purchasing app l on platform $g(j)$ at time t. Next, we describe how we accommodate for customer heterogeneity in the app's demand estimation. As described earlier, the initial distribution of customers taste parameters is

$$\boldsymbol{\alpha}^{p,app}_i = \boldsymbol{\alpha}^{p,app} + \nu^{app}_i \sigma_{app}$$

Also, we know

$$\psi_{i,g(j),l,t} = \psi_{g(j),l,t} + (\boldsymbol{\alpha}^{p,app}_i - \boldsymbol{\alpha}^{p,app})(p_{g(j),l,t}) \tag{35}$$

Similar to the handset demand, the mean *lifetime utility* of an app l as perceived by consumer of type i, $\psi_{i,l,g(j),t}$ is a sufficient statistic for determining if a consumer i on platform $g(j)$ will purchase/download app in period t or not. The share of consumer type i, who will download app l, in period t is given by

$$\hat{s}_{i,l,g(j),t}(\psi_{i,l,g(j),t}) = \frac{\exp(\psi_{i,l,g(j),t})}{\exp(\psi_{i,l,g(j),t}) + \exp(\beta\, \mathbb{E}(W(\psi_{i,l,g(j),t+1}))|\psi_{i,l,g(j),t})))} \tag{36}$$

As before, for every app we only observe aggregate shares. Since, we had approximated the customer distribution using a finite distribution over weights w_i and consumer heterogeneity parameter ν_i, we can write the total share of consumers that download an app l in period t by:

$$\begin{aligned}
\hat{s}_{l,g(j),t}(\psi_{i,l,g(j),t}) &= \sum_i w^{l,g(j)}_{it} \hat{s}_{i,l,g(j),t}(\psi_{i,l,g(j),t}) \\
\sum_i w^{l,g(j)}_{i0} &= 1 \\
w^{l,g(j)}_{it+1} &= w^{l,g(j)}_{it}(1 - s_{ig(j)lt} + q_{it})
\end{aligned}$$

where q_{it} denotes the share of new consumers of type i, who gain access to the platform $g(j)$. The value of q_{it} is used from the handset estimation step (or initialization step). The initial weight parameters $w^{l,g(j)}_{i0}$, are all set to be the same across all apps. However, the future weights and thus the distribution of consumer types can evolve independently for each app across the two platforms. To compute the market shares, we also need access to value functions

$W(\psi)$. Similar to the handset side, we will first estimate the incidental parameters $\phi_{i1}^{app,g(j)}$ and $\phi_{i2}^{app,g(j)}$ using a first stage regression based on current guess of ψ

$$\psi_{i,t+1} = \phi_{i1}^{app,g(j)} + \phi_{i2}^{app,g(j)}\psi_{i,t} + \epsilon_{t+1} \tag{37}$$

We compute the value functions $W(\psi)$ using the imputed values of incidental parameters. To compute the value function, we use a fine grid approximation over ψ and carry out value iteration to estimate W. We next carry out a contraction mapping similar to impute the values of mean utilities from the observed data

$$\psi_{\cdot t}^{h+1} = \psi_{\cdot t}^{h} + \lambda^{app}(s_{\cdot t}^{app}(\psi_{\cdot t}^{h}, \phi^{app}, \sigma^{hs}, \sigma^{app}) - S_{\cdot t}^{app})$$

We add tuning parameter λ^{app} for faster convergence of contraction mapping. We carry out the above iterations until $|\psi_{\cdot t}^{h+1} - \psi_{\cdot t}^{h}|$ is less than the error tolerance threshold. After the convergence on mean utility the incidental parameters are updated using a ordinary least square regression on the updated mean utility. The contraction mapping iterations are again run to get updated mean utility. We stop once we get convergence on the incidental parameters of apps. With the mean utility in hand, we will use the expressions in equation to compute the unobservable product characteristics as a function of model parameters

$$\eta_{jt} = \hat{\psi}_{g(j),l,t} - \vec{x}_{g(j),l,t}\theta^{app}$$

Econometeric error term for the apps model can then simply by constructed by taking the first difference of η. Since we know all time invariant characteristics get cancelled out after taking the first difference, we only include the time varying characteristics in $\vec{x}_{g(j),l,t}^{app}$. Finally, the GMM objective function for the model can be constructed by interacting our econometeric error terms with the specified instruments

$$\arg\min_{\theta^{app}}(\nu_{g(j),l,t}^{app}(\theta^{app})^T \mathbf{Z}_t^{app})\mathbf{W}^{app}(\nu_{g(j),l,t}^{app}(\theta^{app})^T \mathbf{Z}_t^{app})^T$$

where $\mathbf{W}^{app}$ is the weighting matrix. We use $(\mathbf{Z}^{app\prime}\mathbf{Z}^{app})^{-1}$ as our weighting matrix. The model parameters is estimated by search over parameters that minimize the GMM objective function. To estimate the model parameters pertinent to time-invariant characteristics we use a second ordinary least square regression. Finally, we update the σ^{app} parameter, and re-estimate our model. We continue iterating until we find the parameter σ^{app} that minimizes the GMM objective criterion.

3. **Handset Demand** After estimating the app demand model, we get an estimate for $\hat{\Gamma}_i$. Next, we carry out estimation of handset demand parameters in a similar fashion to the initialization phase, with a slight modification.

$$\nu^{hs}_{j,t}(\theta) = \delta_{j,t}(\theta) - \delta_{j,t-1}(\theta) - \boldsymbol{\alpha}^{\boldsymbol{x}}(x_{j,t} - x_{j,t-1}) - \\ \alpha^{\Gamma}(\hat{\Gamma}_{g(j),t}(\cdot) - \hat{\Gamma}_{g(j),t-1}(\cdot)) - \alpha^{p,hs}(\log p_{j,t} - \log p_{j,t-1})$$

Also, the lifetime utility for customer i is given by

$$\delta_{i,j,t} = \delta_{j,t} + (\boldsymbol{\alpha}^{p,hs}_i - \boldsymbol{\alpha}^{p,hs})(\log p_{j,t}) + \alpha^{\Gamma}(\hat{\Gamma}_{g(j),t}(\boldsymbol{\alpha}^{p,app}_i) - \hat{\Gamma}_{g(j),t}(\boldsymbol{\alpha}^{p,app})) \tag{38}$$

We repeat steps 1 and 2, until convergence on model parameters.

A.2. Estimation of Moment Inequalities

We follow a procedure as detailed in Pakes (2012) and Ho and Pakes (2014). Consider the estimator as described in equation (27) i.e.,

$$\hat{\Theta} = \arg\min_{\theta} ||\hat{\Sigma}^{-\frac{1}{2}}_{ii} \min(0, \hat{m}(\theta))|| \tag{39}$$

Since, our moments our independent across the various genres (see Section 4.3), we solve the above optimization problem independentally across the various genres and platforms (for notational ease, we have dropped the platform subscript), i.e.,

$$\hat{\Theta}^g = \arg\min_{\theta} ||{}^g\hat{\Sigma}^{-\frac{1}{2}}_{ii} \min(0, \hat{m}^g(\theta))|| \tag{40}$$

The first step entails estimating the variance covariance matrix to normalize the moments. As it is apparent the sample variance covariance matrix (${}^g\hat{\Sigma}$) is a function of Θ^g. Thus, we adopt the standard two step estimation procedure, just like the estimation of weighting matrix in GMM. In the first step, we estimate Θ^g_0 using an identity matrix as the normalizing factor. Using the estimated set $\hat{\Theta}^g_0$, we then compute ${}^g\hat{\Sigma}$. To compute ${}^g\hat{\Sigma}$, we use centroid of the identified set $\hat{\Theta}^g_0$. Next, we re-estimate Θ^g, using the estimated ${}^g\hat{\Sigma}$ as the normalizing factor.

For set of exogenous functions $h(\cdot) \in \mathbb{R}^2_+$, we use the following specification:

$$h(\Omega_{p,l,t}) = \begin{cases} \mathbb{I}(t = \mathcal{T}_l - k) * |\mathbb{N}(0,1)|, \\ \mathbb{I}(t < \mathcal{T}_l - k) * |\mathbb{N}(0,1)|, \end{cases} \tag{41}$$

where $\mathcal{T}_l - k$ refers to the time when app developer decides to port.

Next, we describe how we estimate the 95% confidence intervals for our set estimator.

1. We start by defining a grid of potential values of porting costs for each genre and platform. We assume a coarse grid uniformly spaced points such that $\Theta_p \in [-30000, 50000]$. We use uniform spacing of 100 for our grid.

2. For each $\theta_p \in \Theta_p$, take 100 random draws from a normal distribution with mean zero and variance-covariance matrix $\mathbb{V}(\hat{\Sigma}(\theta_p)^{-\frac{1}{2}}\hat{m}^g(\theta_p))$. For each random draw, say $\tilde{m}(\theta_p)$, we compute the euclidean norm of its negative part (i.e., compute $||\tilde{m}(\theta_p)_-||$).
3. Compute the 95^{th} quantile of this empirical distribution and let this critical value of our test be $c(\theta_p)$.
4. Now, from the original sample, we compute the actual value of $||\hat{\Sigma}(\theta_p)^{-\frac{1}{2}}\hat{m}^g(\theta_p)||$.
5. If it is less than $c(\theta_p)$, then we accept the value θ_p in the 95% confidence interval, otherwise we reject it.

B. Static Model of Handset Demand

In this Section, we briefly outline the static model of handset demand we used as comparison with our main model in Section 4 of main text. We model the lifetime utility a consumer i gets by buying a handset j at time t as

$$u_{ijt} = \underbrace{x'_{jt}\beta_i + \xi_{jt} + \alpha_i \log p_{jt}}_{\delta_{ijt}} + \epsilon_{ijt}, \tag{42}$$

where $\vec{x}_{j,t}$ refers to the observable characteristics of the handset at time t; $\boldsymbol{\alpha}_i^x$ denotes consumer i's taste towards these characteristics; $\alpha^{p,hs}$ captures consumer's coefficient for price of handset; Finally we have the term $\xi_{j,t}$, that captures the time specific variations from the unobserved mean lifetime utility of app l, which are unobservable to the econometrician but are valued by the consumers;

As before, to allow for heterogeneous preferences among consumers we let

$$\alpha_i = \alpha + \sigma^{hs}\nu_i \quad \nu_i \sim P_\nu(\nu), \tag{43}$$

where P_ν is a standard normal. Σ allows for each component of ν_i to have a different variance. Also let $\theta = (\beta, \alpha, \sigma, \Sigma)$ be a vector containing all the model parameters. Each consumer at time t, chooses to buy/download an app that gives her the maximum lifetime utility. With this assumption the conditional probability that consumer i chooses product j in market t takes the following form:

$$s_{ijt}(\vec{\delta}_t, \theta) = \frac{\exp\left(\delta_{jt} + \log p_{jt}\Sigma\nu_i\right)}{1 + \sum_{k=1}^{k=J}\exp\left(\delta_{kt} + \log p_{kt}\Sigma\nu_i\right)}, \tag{44}$$

such that,

$$\delta_{jt} = x_{jt}\beta + \alpha \log p_{jt} + \xi_{jt}, \tag{45}$$

where N_g refers to the number of apps in genre g. The unconditional choice or aggregate market share of product j at time t is given by:

$$s_{jt}(\delta_t;\theta) = \int s_{ijt}(\cdot)\phi(\nu)d\nu, \tag{46}$$

We numerically approximate the above integral by taking Monte-Carlo draws from P_ν for sufficient number of individuals, such that

$$\hat{s}_{jt}(x_j;\delta,\sigma,\Sigma) \approx \frac{1}{ns}\sum_{i=1}^{ns} \frac{\exp\left(\delta_{jt} + \log p_{jt}\Sigma\nu_i\right)}{1+\sum_{k=1}^{k=J}\exp\left(\delta_{kt} + \log p_{kt}\Sigma\nu_i\right)}, \tag{47}$$

To complete the demand estimation, we first compute observed market shares

$$s_{jt} = q_{jt}/M_t \tag{48}$$

where q_{jt} denotes downloads of app j in time period t, and M_t denotes the market size. For our estimation, we use a market size of 30M. We then equate the observed market shares s_{jt} to the market share imputed in equation (55) such that

$$s_{jt} = \hat{s}_{jt}(x_{\cdot t}, p_{\cdot t}, \delta_{\cdot t}; \sigma). \tag{49}$$

To estimate our model, we use generalized method of moments, similar to our main model. To construct the moment conditions for GMM, we use the unobserved error term $\xi_{jt} - \xi_{jt-1}$, and interact it with a set of instruments Z_{jt}. We use the same set of instruments as our main as described in Section 5.

C. Validity of Instruments

We used a timing assumption to argue for the exclusion of our instrument for the platform utility. The exclusion of our instrument relied on the fact developers might not be able to instantly respond to demand shocks in specific handset sales and hence both current period and one-period lagged value of platform utility can be used as valid instruments. However, it could be speculated that certain developers are able to better track the sales and are quick to respond to shocks in sales of handsets. Thus, to test the robustness of our results, we carry out a 'difference-in-Sargan'[21] test with just one-period lagged value of the platform utility (in specification IV). We are unable to reject the hypothesis that current value of platform utility is excluded from the instrument set. We report our results in Table 14. We also estimate our model with just two-period lagged value of the platform utility in specification V. Our results for both specifications are very similar to those we find in our main analysis.

[21] For more details see Roodman (2007)

Table 14. Estimated Parameters (Handsets)

	Model IV	Model V
$log(prices)$	-2.322**	-2.546***
	(1.034)	(0.719)
ram	0.1457	0.173
	(0.115)	(0.117)
$display\text{-}size$	0.451***	0.415***
	(0.129)	(0.131)
$battery$	0.001***	0.001***
	(0.000)	(0.000)
$log(storage)$	0.489***	0.572***
	(0.047)	(0.047)
$camera$	0.214***	0.219***
	(0.015)	(0.016)
Γ	0.292**	0.296***
	(0.141)	(0.030)
$Handset\ FE$	✓	✓
$\sigma^{p,hs}$	0.008***	0.009***
	(0.002)	(0.002)
N	895	895
GMM Obj	1.17e-4	1.15e-4

Standard errors in parentheses.
Note: *p<0.1; **p<0.05; ***p<0.01

The parameters for time-invariant characteristics are estimating by projecting the estimated handset fixed effects onto the handset time-invariant characteristics using ordinary least square regression.

D. Independence of Apps

As discussed in Section 4, one limitation of our model of demand for apps is – we do not account for substitution across apps. However, as we earlier mentioned extant literature has found that in many products markets, when products are highly differentiated, comprising substitution for dynamics can lead to much better predictive performance. Since, static model allow for the same user base making choice across the entire app portfolio, they inadvertently attribute much higher utility to new apps. To validate this, we estimate a static model of demand in a spirit similar to Berry et al. (1995) and Ghose et al. (2012).

To compare the out-of-sample prediction performance of both the models, we first randomly sample 1500 apps from both Apple App Store and Google Play. We make sure all apps existed in the first two quarters of 2013 (i.e., January 2013 – June 2013), and also in the last two quarters of 2014 (i.e., July 2014 – December 2014). Next, we estimate both a static model and a dynamic model on the first half of the data i.e., only using the data from the first two quarters of 2013.

Table 15. Out-of-Sample MSE for Static vs Dynamic Model of Apps

	iOS	Google-Play
Static	1066842.3	5636.1
Dynamic	714.5	1369.0

Note. The Table compares the out-of-sample prediction performance of a static model of app demand, that accounts for substitution across apps and a dynamic model of demand that compromises substitution to account for consumer dynamics. We first randomly sample 1500 apps from both Apple App Store and Google Play. We make sure all apps existed in the first two quarters of 2013 (i.e., January 2013 – June 2013), and also in the last two quarters of 2014 (i.e., July 2014 – December 2014). Next, we estimate both a static model and a dynamic model on the first half of the data i.e., only using the data from the first two quarters of 2013. Next, using these estimates we make predictions on the latter half of the data i.e., last two quarters of 2014.

Next, using these estimates we make predictions on the latter half of the data i.e., last two quarters of 2014. In Table 15, we report the mean squared error of both the methods on the second half of the data. We find, that our dynamic model of demand provides much lower MSE, and hence, better predictions than a static model.

For completeness, we briefly outline the static model of demand for apps. We advise the readers to refer to Berry et al. (1995) for further details. We model the lifetime utility a consumer i gets by downloading an app j at time t from platform $g(j)$ as

$$u_{ijt} = \underbrace{x'_{jt}\beta_i + \eta_{jt} + \alpha_i p_{jt}}_{\psi_{ijt}} + \epsilon_{ijt}, \tag{50}$$

where $\vec{x}_{j,t}$ refers to the observable characteristics of the app at time t; $\boldsymbol{\alpha}_i^x$ denotes consumer i's taste towards these characteristics; $\alpha^{p,app}$ captures consumer's coefficient for price of app; Finally we have the term $\eta_{j,t}$, that captures the time specific variations from the unobserved mean lifetime utility of app l, which are unobservable to the econometrician but are valued by the consumers;

As before, to allow for heterogeneous preferences among consumers we let

$$\alpha_i = \alpha + \sigma^{hs}\nu_i \quad \nu_i \sim P_\nu(\nu), \tag{51}$$

where P_ν is a standard normal. Σ allows for each component of ν_i to have a different variance. Also let $\theta = (\beta, \alpha, \sigma, \Sigma)$ be a vector containing all the model parameters. Each consumer at time t, chooses to buy/download an app that gives her the maximum lifetime utility. With this assumption the conditional probability that consumer i chooses product j in market t takes the following form:

$$s_{ijt}(\vec{\delta}_t, \theta) = \frac{\exp\left(\delta_{jt} + \log p_{jt}\Sigma\nu_i\right)}{1 + \sum_{k=1}^{k=J}\exp\left(\delta_{kt} + \log p_{kt}\Sigma\nu_i\right)}, \tag{52}$$

such that,

$$\delta_{jt} = x_{jt}\beta + \alpha p_{jt} + \xi_{jt}, \tag{53}$$

where N_g refers to the number of apps in genre g. The unconditional choice or aggregate market share of product j at time t is given by:

$$s_{jt}(\delta_t;\theta) = \int s_{ijt}(\cdot)\phi(\nu)d\nu, \tag{54}$$

We numerically approximate the above integral by taking Monte-Carlo draws from P_ν for sufficient number of individuals, such that

$$\hat{s}_{jt}(x_j;\delta,\sigma,\Sigma) \approx \frac{1}{ns}\sum_{i=1}^{ns}\frac{\exp\left(\delta_{jt} + \log p_{jt}\Sigma\nu_i\right)}{1 + \sum_{k=1}^{k=J}\exp\left(\delta_{kt} + \log p_{kt}\Sigma\nu_i\right)}, \tag{55}$$

To complete the demand estimation, we first compute observed market shares

$$s_{jt} = q_{jt}/M_t \tag{56}$$

where q_{jt} denotes downloads of app j in time period t, and M_t denotes the market size. For our estimation, we use a market size of 30M. We then equate the observed market shares s_{jt} to the market share imputed in equation (55) such that

$$s_{jt} = \hat{s}_{jt}(x_{\cdot t}, p_{\cdot t}, \delta_{\cdot t};\sigma). \tag{57}$$

To estimate our model, we use generalized method of moments, similar to our main model. To construct the moment conditions for GMM, we use the unobserved error term $\eta_{jt} - \eta_{jt-1}$, and interact it with a set of instruments Z_{jt}. We use the same set of instruments as our main as described in Section 5.

E. Estimating App Downloads

We build on methods introduced by Garg and Telang (2013), to estimate app sales from app download rank data. Our data source contains daily download ranks across -63 and -56 app categories for iOS and Google Play respectively. However, we have multiple missing entries in the overall sales rank. This is mostly due to products having very low downloads and thus very high overall download rank. To address the missing download rank issue, we use rank in other categories to impute the overall sales rank.

$$D_{c_1} = D_{c_2} \tag{58}$$

To estimate missing ranks in overall rank for apps, we use subcategory rankings and project them onto the overall rank. This procedure leaves us still with 3% and 4% missing ranks in iOS and

Table 16. Handset Characteristics Description.

Data	Description
Display Size	The screen size varies from model to model and is the diagonal length of screen measured in inches.
Primary Camera	Smartphones have built in digital cameras, and we measure it through their resolution in Megapixels
RAM	RAM is the random access memory of the handsets and is measured in MB.
Storage	This is physical storage available in the phones, measured in GB.
Battery	Battery is measured in milliamp hours mAh.

Table 17. App Genre Descriptives

	iOS	Google Play
Games	42%	63%
Entertainment	7%	4%
Education	4%	2%
Lifestyle	5%	2%
Music	4%	3%
Photo and Video	8%	3%
Social Networking	4%	3%
Utilities	6%	7%
Others	19%	13%

Note. The genre statistics are calculated across our entire sample of 13,731 iOS and 6,950 Google Play apps, respectively.

Google Play respectively. We use linear interpolation to impute the missing rank data. Next, we use the pareto distribution to estimate app downloads as a function of download rank. To calibrate the parameters for the pareto distribution we source download data for 300 iOS and 300 Google Play apps from a market research firm that tracks app downloads in the united states. For a detailed discussion, we advise readers to refer to Garg and Telang (2013).

F. Test of Model Misspecification

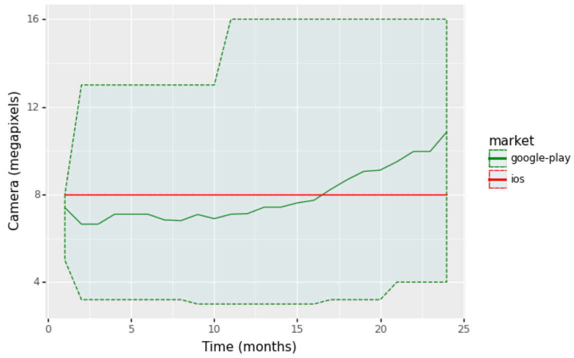
Ho and Pakes (2014) describe a procedure to test if the moment inequalities are correctly specified. We follow a similar procedure as theirs. Given, we estimate non-empty confidence interval sets we can not reject the null that the model is misspecified.

Table 18. Smartphones with Highest Observed Market Share

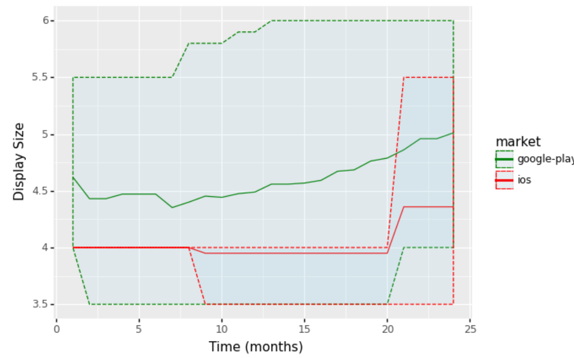
		(Market Share)
Apple	Apple iPhone 5S	0.13
	Apple iPhone 5C	0.08
	Apple iPhone 5	0.06
	Apple iPhone 6	0.05
Android	Samsung Galaxy S4	0.10
	Samsung Galaxy S5	0.05
	Samsung Galaxy Note 3	0.04
	Samsung Galaxy S3	0.04

Note. The table reports the top smartphones with highest observed market shares for the period of our study.

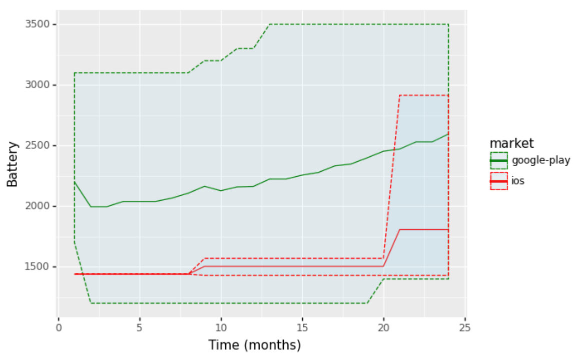
Figure 19. Handset Characteristics Evolution (Nonstandardized)



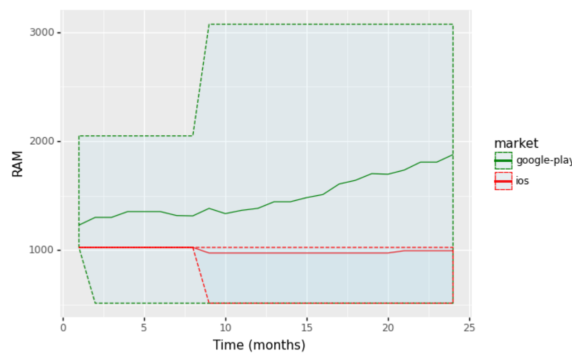
(a) Camera (Megapixels)



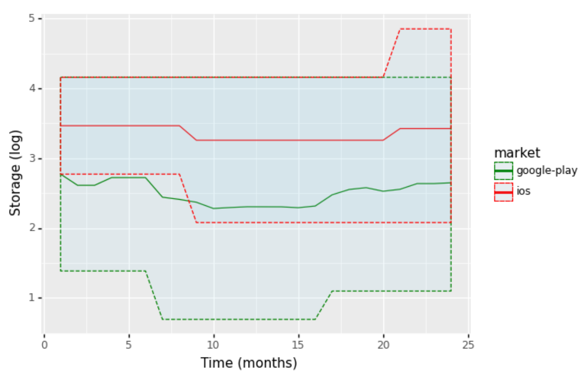
(b) Display Size



(c) Battery (mAh)



(d) RAM



(e) Storage (log(GB))

Note. The figures report how the handset features for the entire mix of handsets in the market evolve over time. The solid line denotes the average value of handset characteristics over time and the dashed line indicate the minimum and maximum value of handset characteristics over time. Time period 1 denotes January 2013 and Time period 24 denotes December 2014.