

Bayesian Estimation of a Dynamic Equilibrium Model of Pricing and Entry in Two-Sided Markets: Application to Video Games*

Yiyi Zhou[†]

October 14, 2011

(Job Market Paper)

Abstract

This paper empirically studies the impacts of pricing choices by platform intermediaries in two-sided market with positive indirect network effects. It presents a dynamic equilibrium model to analyze consumers' purchase decision for competing hardware platforms and their affiliated software products, and software firms' dynamic pricing and entry decisions. I provide a new practical methodology for structural estimation of dynamic equilibrium models. It combines the Bayesian Markov Chain Monte Carlo algorithm and the dynamic equilibrium model solution algorithm into a single algorithm that estimates the parameters and solves the model simultaneously. I apply it to U.S. fifth-generation video game industry (May 1995 - February 2002). Results suggest that overpricing one side of the market not only discourages demand on that side but also discourages participation on the other side, and so can lead to a death spiral.

Keywords: Bayesian Estimation, Markov Chain Monte Carlo, Dynamic Equilibrium Model, Two-Sided Market, Indirect Network Effect, Dynamic Pricing and Entry

*I am extremely indebted to my advisers, Steven Stern and Simon Anderson, for their numerous advice, guidance and support. I am also thankful to my third reader, Federico Ciliberto, for his comments and suggestions. I also benefit a lot from the conversations with Ambarish Chandra, Yongmin Chen, Andrew Ching, Leora Friedberg, Nathan Larson, Toshihiko Mukoyama, Christopher Otrok, John Pepper, Regis Renault, John Rust, Eric Young, Glen Weyl, Kenneth Wolpin and Daniel Xu. Furthermore, I thank Wei-Min Hu for assistance in acquiring data. All errors in the paper are my own.

[†]Ph.D. Candidate in Economics at University of Virginia. Email: yz6x@virginia.edu

1 Introduction

In many two-sided or platform markets, consumers join a platform to access goods provided by firms who are affiliated with that platform, and firms join a platform to reach consumers who have joined that platform. The number of consumers on a platform depends on the availability, quality and prices of the affiliated products, and the success of the affiliated products depends on the number of consumers. This kind of interdependence or externality between groups of agents that the platform intermediary serves is called indirect network effects in the literature on two-sided markets. Moreover, platform markets are often inherently dynamic environments due to the durability of platform intermediaries and the affiliated products. In the dynamic two-sided market environment, overpricing one side of the market not only discourages demand on that side but also discourages participation on the other side, and so can lead to a death spiral.

This paper presents a dynamic equilibrium model to analyze consumers' purchase decision for competing hardware platforms and their affiliated software products, and software firms' dynamic pricing and entry decisions. On the one side of the market, consumers are heterogeneous and forward-looking, and have rational expectations about software entry and prices in the future. They choose whether to purchase the hardware and the affiliated software. The hardware purchase decision and the software purchase decision are connected with each other. On the one hand, since hardware usually has little stand-alone value, hardware value comes from the total utility from being able to purchase the affiliated software products. Hence, consumers rationally anticipate the software market when they make their purchase decisions of hardware at first place. On the other hand, since only those consumers who have joined a hardware platform are potential buyers for the affiliated software products, the number of active consumers for a software product depends on how many consumers have purchased the compatible hardware.

On the other side of the market, there are finite separate software submarkets.¹ In each submarket, incumbents decide how much to charge and potential entrants decide whether to enter. They make their choices in a dynamic environment due to four reasons. Firstly, the consumption value of a software product decays over time because consumers favor new products. Secondly, the competition environment changes over time, mainly driven by the entry of new software products. Thirdly, the distribution of potential buyers in a submarket changes over time. Heterogeneous con-

¹In the context of videogames, I define a software submarket that a game title belongs to based on the console that game is compatible with and the game genre it is grouped in. For example, Football games on PlayStation 1 is a submarket, Baseball games on PlayStation 1 is a submarket, Football game on Nintendo 64 is another submarket, and so on.

sumers purchase hardware or enter the affiliated software submarkets at different point of time, and they also purchase software and leave a software submarket at different point of time. This gives software firms an incentive to engage in inter-temporal price discrimination. Lastly, consumers are forward-looking. If prices are declining and the product variety is growing over time, consumers have an option value associated with waiting. Firms also take into account how their price and entry actions affect the option value from waiting.

In dynamic equilibrium models, given other agents' strategies, each agent's best response is itself the solution to single-agent dynamic programming problem; moreover, the equilibrium is the fixed point of the system of best response operators. For relatively complicated models, it is extremely difficult and even impossible to compute the continuation value and policy function. This paper provides a new practical methodology for structural estimation of dynamic equilibrium models. It combines the Bayesian Markov Chain Monte Carlo (MCMC) algorithm and the dynamic equilibrium model solution algorithm into a single algorithm that estimates the parameters and solves the model simultaneously. It substantially reduces the computation burden and makes it possible to estimate complicated dynamic equilibrium models.

For a particular point of the state space, I employ a nonparametric approximation method to obtain the perception of other agents' equilibrium strategies, and the value function given that other agents play the perceived equilibrium strategies. To be more specific, I partially solve for each agent's best response and value functions given other agents play the perceived equilibrium strategies at each draw of parameter vectors, and use those pseudo-best response functions and pseudo-value functions to non-parametrically approximate the perceived equilibrium strategies and the value functions at the current trial parameter vector.

However, it is still computationally burdensome and even impossible to obtain continuation values which requires to compute the next-period value functions at all possible points of the state space. Hence, I modify the simulation and interpolation method proposed by Keane and Wolpin. In the first step, I randomly choose a subset of the state points every period, and obtain the values at those points with the non-parametric approximation approach. Then, I interpolate the value functions with a quadratic-in-states polynomial approximation. Lastly, for each current state, I simulate a next-period-state using the non-parametrically approximated equilibrium strategies, and then use the predicted value at that simulated next-period-state as the continuation value.

Furthermore, it is possible that the proposed model has more than one equilibrium. The existence of multiple equilibria can cause serious problems for empirical studies. Models with multiple

equilibria do not have a unique reduced form. To accommodate the possibility of multiple equilibria, I adopt a stochastic equilibrium selection mechanism which assigns an exogenous probability to each possible equilibrium. I sample those probabilities and the parameters of the model jointly from the posterior distribution.

The estimation method is applied to U.S. fifth-generation video game industry (May 1995 - February 2002). Counterfactual experiments suggest that Sega Saturn got wrong the two-sided business pricing model and hence was shaken out of the market. It would have survived by using one (or more) of the following strategies: (a) reducing its console price to attract more consumers and hence more games to come on board; (b) subsidizing software R&D to encourage more games and more consumers to participate; or (c) lowering royalty fees to reduce the game price and attract more consumers and more games.

The rest of the paper is organized as follows. In the remainder of this section I provide a brief review of the related literature. In section 2, I describe the data set and the U.S. video game industry. In section 3, I build a structural equilibrium model with dynamic demand and dynamic supply. In section 4, I propose the Bayesian MCMC estimation algorithm and discuss the related computational issues. In section 5, I report the estimation results and the fit of the model. In section 6, I conduct three counterfactual exercises to analyze the impacts of platform's price choices. Section 7 concludes.

Related Literature

This paper contributes to the literature on two-sided markets. Most previous literature on two-sided markets are theoretical studies, for example, Armstrong (2006), Rochet and Tirole (2003), Hagiu (2006), Weyl (2010). The main result is that pricing on one side of the market depends not only on the demand on that side but also on how it affect participation on the other side. However, few empirical results exist. This paper adds to empirical studies on platform pricing in two-sided markets.

On the demand side, this paper is related to the literature on estimating dynamic models of demand for complementary durable goods, including Lee (2010) and Gowrisankaran, Park and Rysman (2011). Both of them restrict the dynamic behavior of consumers by assuming that consumers condition their decision to purchase any product only on the current realization of a scalar state variable that is a sufficient statistic for distribution of expected payoffs. In other words, it is assumed that the maximum expected payoff that consumers can get from participating in the market

is Markovian. As already recognized by others (e.g. Hendel and Nevo 2007), consumers are expected to condition their behavior on all the variables that affect firm behavior. Therefore, such assumption is difficult to reconcile with a general supply model, in which firms condition their actions on the actions of each individual competitor and consumer responses. In contrast, consumers in this paper are assumed to be forward-looking and have rational expectations on future environment. This assumption on consumers behavior is always adopted in theory literature and can be reconciled with a consistent supply model.

On the supply side, this paper is related to the literature on dynamic pricing and the literature on dynamic entry. Most of the previous papers on dynamic pricing in the durable goods markets are theoretical studies which have focused on establishing closed-form results for relatively simple model, but few results exist for the more complicated multi-firm, multi-characteristic settings that are common in the empirical literature on differentiated products markets. One exception is Nair (2007), who considers the problem of a video game seller as a single product durable-goods monopolist. In the literature on empirically estimating dynamic entry models, almost all of the previous studies have described the demand side in a static way, for example, Ericson and Pakes (1995), and Bajari, Benkard and Levin (2007). To my knowledge, no previous research paper has focused on both dynamic pricing and dynamic entry.

My estimation method is related to the Imai, Jain and Ching (2009) which proposes Bayesian method for estimating single-agent dynamic discrete choice models. The IJC algorithm uses the Metropolis-Hastings algorithm to draw a sequence of parameter vectors from their posterior distributions. During each MCMC iteration, it partially solves for the value functions at each draw of parameter vectors, stores those partially solved value functions (pseudo-value functions) and uses them to non-parametrically approximate the expected value functions at the current trial parameter vector.²

My estimation method is also related to the empirical literature on estimating dynamic equilibrium models. Compared to single-agent dynamic programming models, dynamic equilibrium models have another layer of complication in the sense that the equilibrium of the dynamic game is the fixed point of the best response system. The most popular methods in the literature is the nested fixed point approach (e.g. Berry, Levinsohn, and Pakes 1995) and two-step approaches (e.g., Bajari, Benkard, and Levin 2007). The NFP approach requires to solve for the equilibrium for each guess of

²They also show that as the MCMC iterations and the number of past pseudo-value functions for approximating the expected value functions increase, pseudo-value function will converge to the true value functions and the posterior parameters draws based on the pseudo-value functions will also converge to the true posterior distribution.

parameter vector and thus can not handle complicated models. The two-step approaches sidestep the equilibrium computation step by substituting nonparametric functions of the data for the policy functions but can not easily deal with unobserved state variable problems. Contrast to the existing approaches, the Bayesian MCMC method proposed by this paper combines the algorithm of partially solving for each agent’s optimization problem with the non-parametric approximation algorithm.

The estimation method proposed by this paper also contributes to the empirical literature on multiple equilibria problems. Classical estimation methods such as maximum likelihood and other extreme estimators require that we obtain all the equilibria for every trial value of the parameters. This can be infeasible even for simple models. Ciliberto and Tamer (2009) notice that a model may not make exact predictions about outcomes, it still may restrict the range of possible outcomes. They focus on oligopoly entry models and estimate bounds on the demand and cost parameters of firms’ profits. Bajari, Hong and Ryan (2010) suggest and estimate an equilibrium selection mechanism.³ In the spirit of their method, I also adopt a stochastic equilibrium selection mechanism. I assign each possible equilibrium with an exogenous probability, which is similar to Tamer (2003) that let the probability of each equilibria depend on the exogenous x ’s observed in the data. In the process of Bayesian MCMC algorithm, I treat those equilibrium probabilities the same as parameters, and sample the parameters and those equilibrium probabilities jointly from the posterior.

2 The U.S. Videogame Industry

Since the early 1970’s when Pong was introduced, the U.S. video game industry has grown to reach 22 billion dollars in revenue in 2008, over twice the total box-office revenue in the movie industry (10 billion dollars). The video game industry is a two-sided market in which console hardware act as platform intermediaries, and consumers and software producers are on the two sides of the market. On the one hand, console providers design and sell consoles to consumers who pay one-time fixed fee to join a platform (i.e., console price). On the other hand, they charge game producers a royalty fee for the rights to the code which allows the game producers to make their games compatible with the console. The royalty fee is not one-time payment, rather, fixed payment per copy sold to consumers. To track sales for royalty collection, console providers actually manufacture all video

³They propose four criteria for equilibrium selection: (i) whether the equilibrium involves mixed strategies; (ii) whether the equilibrium is efficient, in the sense that it maximizes joint payoffs; (iii) whether the equilibrium is Pareto dominated; and (iv) whether the equilibrium is self-reinforcing based on the Nash product of a agent’s utilities for pure strategy equilibria.

games themselves to ensure control over the printing process.

To satisfy the needs of consumers for the latest technology, console providers have introduced new systems approximately every five years. The fifth-generation was dominated by three consoles, Sega Saturn released in May 1995, Sony PlayStation One released in September 1995, and Nintendo 64 released in September 1996.

2.1 Data

The main data is obtained from NPD Group, a market research firm. It includes the monthly revenue and the unit sales of the three fifth-generation consoles, Sega Saturn, Sony PlayStation One and Nintendo 64, from May 1995 until February 2002. The console price is calculated by taking the ratio of the revenue over the unit sales in each month. Since the sixth-generation started when Sony launched its PlayStation 2 in October 2000, the data covers the entire fifth-generation video game industry.

It also includes the monthly revenue and the unit sales for 1697 unique software titles released for the three consoles during the sample period. It is collected from approximately thirty of the largest retailers in the U.S., which account for approximately 85% of video game sales, and is extrapolated by NPD for the entire U.S. market. Similarly, the software price is calculated by taking the ratio of the revenue over the unit sales in every month. The data that I use for estimating the software market only includes sports games. The main reason is that it is relatively easy to sort sports games into groups and it also reduces the estimation time to use a smaller sample. The data for estimation contains 397 products distributed in 29 submarkets. I also collect the data of the critics and user rating score for each game title from several largest websites such as IGN, gamerankings, GameSpot and Gamasutra.

General descriptive statistics are provided in table 1. Up to February 2002, the installed bases of users in U.S. market for the Sega Saturn, PlayStation One and Nintendo 64 are 1.28 millions, 28.25 millions and 17.17 millions, and the total unit sales of their affiliated software products are 8.09 millions, 300.02 millions and 111.55 millions, respectively. Even though Sega Saturn was the first mover, it became the “other” system barely two years after its release, running a distant third behind its two main competitors.

Table 1: Statistics of the U.S. Fifth-generation Video Game Industry

	Sega Saturn	PlayStation	Nintendo 64
HARDWARE			
Release Date	May 1995	Sept. 1995	Sept. 1996
Provider	Sega	Sony	Nintendo
CPU bits	32	32	64
MHZ	28	33.87	93.75
Starting price	\$399.9	\$299.7	\$199.8
Ending price	\$41.0	\$112.2	\$87.1
Average unit sales per month (million)	0.02	0.36	0.26
Installed base (Feb. 2002, million)	1.28	28.25	17.17
SOFTWARE			
Total active titles	240	1172	385
Total Unit Sales (Feb. 2002, million)	8.09	300.20	111.55
Average starting price (\$)	\$52.66	\$41.57	\$54.57
Average starting units sales (thousand)	3.12	21.72	45.26
Average units sales per title (thousand)	33.72	256.14	289.74

What happened to Sega Saturn? Two main factors contribute to its failure. First, it was released at a high price of \$399, \$100 higher than PlayStation (\$299) and \$200 higher than Nintendo 64 (\$199). In two-sided markets, high price on consumer side discourages the entry of consumers, and hence the entry of software producers whose goal is to reach the consumers who have joined the platform. Especially when there are multiple competing platforms, the effect of participation of one side on the other has even more bite. High console price lose consumers to the competing platforms, which upgrades the value of the competitors to buyers, and hence leads to a large decrease in buyer interest in the original platform. Besides, it was widely agreed that Saturn was difficult to develop for compared to PlayStation, due to its dual-CPU architecture and its more complex graphics.⁴

2.2 Industrial Description

Below I briefly discuss three important features regarding this industry.

1. Positive Indirect Network Effect

In the video game industry, consumers join a platform to access video games that have been released for that platform, and game producers enter a platform to reach consumers who have joined that

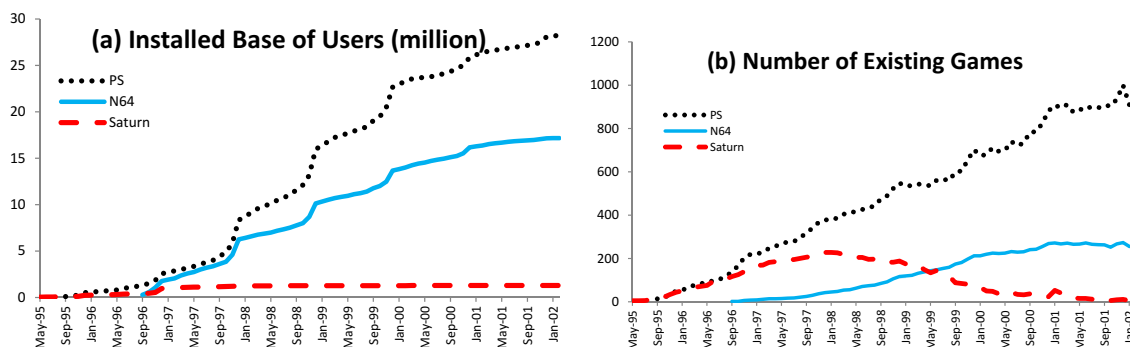
⁴“One very fast central processor would be preferable. I don’t think all programmers have the ability to program two CPUs—most can only get about one-and-a-half times the speed you can get from one SH-2. I think that only 1 in 100 programmers are good enough to get this kind of speed [nearly double] out of the Saturn. ” —Yu Suzuki reflecting upon Saturn Virtua Fighter development. Wikipedia provides more details about the difficulties of developing games for Saturn, see http://en.wikipedia.org/wiki/Sega_Saturn#cite_note-16.

platform. Hence, the installed base of users of a console is largely contingent on the availability and prices of the games that have been or expected to be released for that console, and the entry of games to a console largely depends on how many users have purchased or expected to purchase that console.

On the one side of the market, consumers make their purchase decisions of both hardware and software products. Hardware usually has little if any stand-alone value. Its value comes from the compatible software titles that either have been or are expected to be released. Figure 1 (a) presents the installed base of users of the three consoles in the sample period. The installed bases of Sony PlayStation and Nintendo 64 were growing fast during this period, because consumers keep on purchasing these two consoles. On the contrary, the consumer network of Sega Saturn stopped growing one and half year after it was released.

On the other side of the market, incumbent software firms set their prices and potential entrants decide whether to enter the market. Figure 1 (b) presents the number of existing software titles sold for each console in every month during the sample period. The software varieties of Sony PlayStation and Nintendo 64 were growing fast, because new software products keep on entering these two platforms. On the contrary, the software variety of Sega Saturn started to shrink from January 1998, because no new software products join that platform.

Figure 1: Positive Indirect Network Effect

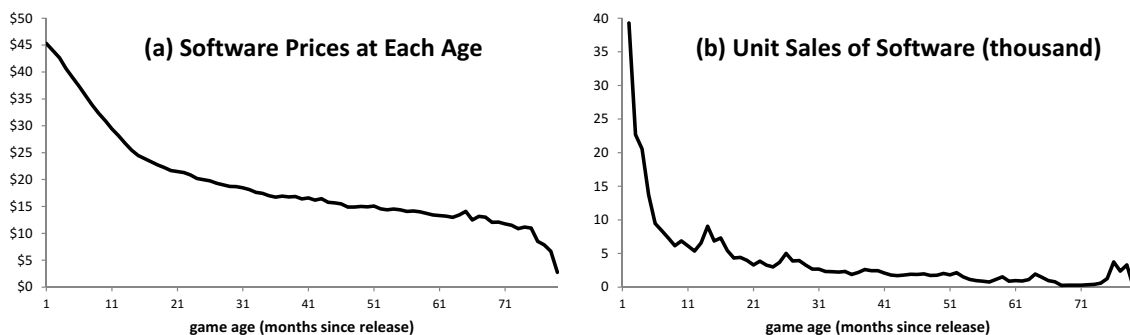


2. Software Prices and Sales Declines with Age

The second feature of the U.S. video game market is that the software price and unit sales start at a high level and then decline as the product ages, especially in the first six months after it is released. In figure 2, the horizontal axis is the software age measured by the months since introduction and the vertical axis is the average software price for (a) and the average unit sales for (b). The software

price is around \$46 per copy when it is released and then drops to half of the initial price after one year. The price declines by 21% on average in the first six months. The average software unit sales is around 40 thousand per copy in the first month and then falls to around 5 thousand after one year.

Figure 2: Prices and Unit Sales of Video Games at Each Age



Note: Monthly data of 1697 games released for Sony PlayStation One, Nintendo 64 and Sega Saturn from May 1995 to February 2002.

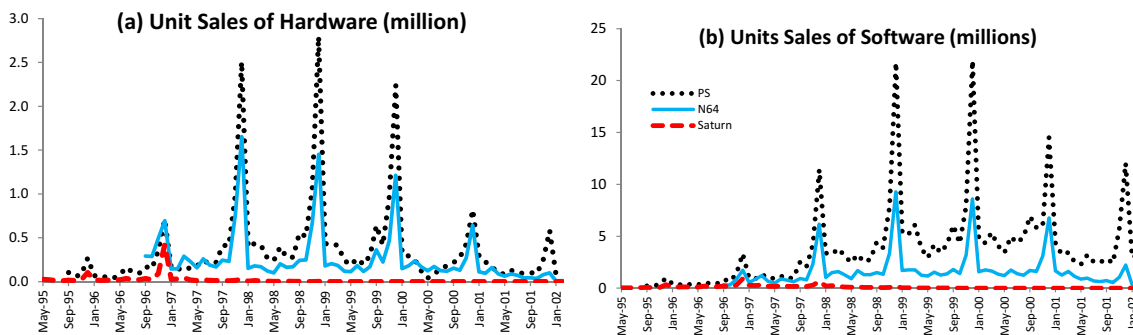
What drives the software price and unit sales to drop so quickly? A falling-cost explanation is not convincing for this industry. Once a software title is developed, software producers only need to pay royalty fees to console makers and production/package costs. Both costs remain roughly constant per unit over time.⁵ Inter-temporal price discrimination seems to be a good argument for the declining pattern of prices and sales. Consumers are heterogeneous in terms of their preferences to either product characteristics or price or both. Hence, they make their purchases of consoles and games at different point of time. As a result, the distribution of potential buyers of games changes over time. The different composition of consumers at different point of time induces firms to charge different prices. Intuitively, consumers with high net-valuations make their purchases earlier than those with low net-valuations. Thus, it is optimal for firms to set high initial prices to sell to consumers with high net-valuations and cut prices thereafter to appeal to those with low net-valuations. This partly explains the decreasing pattern of software prices and sales. Besides, the entry of new software products leads to more intensive competition and thus forces the existing products to cut their prices.

⁵Coughlan (2001) reports that production/package costs for 32-bit CD-ROM games remains roughly constant at \$1.5 per disc. Nair (2007) reports that the royalty fee for the 32-bit Sony PlayStation compatible games was pre-announced and held fixed at \$10 by Sony throughout the life-cycle.

3. Seasonality

Figure 3 shows the monthly unit sales of each console and the monthly unit sales of affiliated games from May 1995 till February 2002. During holiday months (November and December) sales are easily double or triple the average sales in other months.

Figure 3: Seasonality



3 Model Framework

In this section, I present a dynamic equilibrium model of demand and supply. The model is dynamic, time is discrete and the horizon is finite. The following events occur in each time period.

The software market consists of M separate submarkets. In each submarket, there are two groups of software firms: incumbents and potential entrants. Incumbent firms decide how much to charge, and potential entrants decide whether to enter. Price and entry decisions are made simultaneously.

Consumers immediately observe the software prices. Consumers who have not owned any hardware decide whether to purchase a hardware and, if so, which one. Once they join a hardware platform, they start to buy the affiliated software products. Consumers who have owned a hardware decide whether to make a purchase of affiliated software and, if so, which software to purchase.

Software Entry decisions are implemented. New entrants are carried out and their uncertain outcomes are realized. We move to the next period.

Below, I first describe consumer dynamic purchase of hardware and software, software producers' dynamic pricing and entry behavior, and lastly give the equilibrium definition of the model.

3.1 Demand For Hardware

Allowing consumer heterogeneity is important in the model. It explains why consumers join a platform and purchase hardware and software at different point of time. As a result, the distribution of active consumers in the software market changes over time. This generates an incentive for software firms to engage in inter-temporal price discrimination. I assume a discrete finite number of consumer types in the population (indexed by i), each having the same preference for product characteristics but with different preferences over price.

I assume that each consumer can only purchase at most one hardware.⁶ The lifetime expected utility of consumer type i from purchasing platform l at time t is

$$U_{ilt} = \Gamma_{ilt} + X_{it}^h \psi_h - \varphi_i p_{it}^h + \xi_{it}^h + \epsilon_{ilt}^h, \quad (1)$$

where Γ_{ilt} is the expected value of being able to purchase software associated with platform l . The functional form is derived from the software adoption portion of the model, which will be described in the next subsection. X_{it}^h is a vector of observed hardware product characteristics. It includes the platform-specific fixed effects, and the holiday dummy which equals to 1 if it is November or December and equals to zero otherwise. p_{it}^h is the hardware price which treated as exogenous in this paper.⁷ ξ_{it}^h is additional hardware characteristics observed by consumers but not by researchers. ϵ_{ilt}^h is idiosyncratic consumer taste. ψ_h represents consumers preferences in observed hardware characteristics, and φ_i represents consumer type-specific sensitivity to price.

Since hardware products are durable goods, consumers are forward-looking when they decide whether to buy them. No-purchase captures the option value of delaying purchases to a future period. I specify the utility of not buying at time t as the sum of the discounted expected value of waiting and an idiosyncratic consumer taste:

$$U_{i0t} = \beta_c E_t \left[\max\{\max_l U_{ilt+1}, U_{i0t+1}\} \right] + \epsilon_{i0t}^h, \quad (2)$$

where β_c is the consumer's discount factor and the expectation is taken with respect to the distribu-

⁶Ruling out multiple console purchasing may potentially cause biases. This paper does not allow for consumer multihoming for two main reasons. First, including multihoming purchase significantly complicates the estimation. Lee (2010) allows for multihoming but he does not model the supply side. However, the model in this paper is an equilibrium model of both demand and supply. Second, precise data on the degree of multihoming is unavailable.

⁷It is widely speculated that all the major consoles were sold at a price near marginal cost. The literature on two-sided markets, e.g., Armstrong (2006) and Hagiu (2006), has provided good reasons why it is in the interest of a platform provider to keep the price of one side low and make money from the other side. Hence, we can think that the declining console price is mainly driven by the declining costs of producing consoles.

tion of future variables unknown to the consumer conditional on the current information. As usual in the literature, ϵ_{ilt}^h and ϵ_{i0t}^h are assumed to follow the standard Type-I Extreme Value distribution and are *i.i.d.* over time, products and consumer types.

Let \mathbf{S}_t denote the information set that affect consumer purchase decision of hardware at time t . Then, a type- i -consumer's dynamic optimization problem can be written as

$$H_i(\epsilon_{it}^h, \mathbf{S}_t) = \max \left\{ \max_l U_{ilt}, \epsilon_{i0t}^h + \beta_c E \left[E_{\epsilon^h} H_i(\epsilon_{it+1}^h, \mathbf{S}_{t+1}) \mid \epsilon_{it}^h, \mathbf{S}_t \right] \right\},$$

where $H_i(\epsilon_{it}^h, \mathbf{S}_t)$ is consumer type- i 's value function with information set \mathbf{S}_t and tastes ϵ_{mit} . Let $H_i(\mathbf{S}_t)$ denote the expected value function, that is, the value function before consumers know their demand shocks ϵ_{it}^h :

$$H_i(\mathbf{S}_t) = \int_{\epsilon_{it}^h} H_i(\epsilon_{it}^h, \mathbf{S}_t) dF_{\epsilon^h}(\epsilon_{it}^h)$$

Following Rust (1987), the integration with respect to the extreme value error terms has a closed form, and the deterministic component of the consumer's value function satisfies

$$H_i(\mathbf{S}_t) = \log \left\{ \sum_l \exp(\Gamma_{lt} + X_{lt}^h \psi_h - \varphi_i p_{lt}^h + \xi_{lt}^h) + \exp[\beta_c E H_i(\mathbf{S}_{t+1} \mid \mathbf{S}_t)] \right\}.$$

Then, the probability that a type i consumer purchases hardware l at time t is

$$b_{il}^h(\mathbf{S}_t) = \frac{\exp(\Gamma_{lt} + X_{lt}^h \psi_h - \varphi_i p_{lt}^h + \xi_{lt}^h)}{\exp[\beta_c E H_i(\mathbf{S}_{t+1} \mid \mathbf{S}_t)] + \sum_k \exp(\Gamma_{kt} + X_{kt}^h \psi_h - \varphi_i p_{kt}^h + \xi_{kt}^h)}.$$

Hence, the demand for the hardware l in period t is $q_{lt}^h = \sum_i n_{it}^h b_{il}^h$, where n_{it}^h is the number of consumers who have not purchased any hardware at time t . In dynamic models of discrete choice demand, $\{n_{it}^h\}_{t=1}^T$ evolves as

$$n_{it+1}^h = n_{it}^h (1 - \sum_l b_{il}^h).$$

3.2 Demand For Software

I assume that the software market consists of M separate submarkets, explicitly ruling out competition across submarkets. Each consumer can only purchase at most one software product within a submarket, explicitly allowing for competition within a submarket. In the video game industry, a submarket is defined as the game genre and the compatible console. In Appendix B, I find supporting evidence for this assumption as sports video games are found to be strong substitutes within a

submarket and weak substitutes across submarkets empirically.

Software Utility

Let J_{mt} denote the set of software products available for consumers to purchase in submarket m at time t . The lifetime utility that consumer type i can get from purchasing a software product $j \in J_{mt}$ at time t is given by

$$u_{ijt} = X_{jt}\psi - \varphi_i p_{jt} + \xi_{jt} + \epsilon_{ijt}, \quad (3)$$

where X_{jt} is a vector of observed software product characteristics, including the online rating score, product newness and holiday dummy; p_{jt} is the software product price; ξ_{jt} is additional software characteristics observed by consumers but not by researchers; and ϵ_{ijt} is idiosyncratic consumer taste. Here, ψ represents consumers preferences in observed software characteristics and φ_i represents consumer type-specific sensitivity to software price.

Because consumers are forward-looking, no-purchase captures the option value of delaying purchases to a future period. I specify the utility of not buying in the submarket m at time t as the sum of the discounted expected value of waiting and an idiosyncratic consumer taste:

$$u_{im_0t} = \beta_c E_t \left[\max_{j \in J_{m_{t+1}}} \{ \max_{j \in J_{m_{t+1}}} u_{ij_{t+1}}, u_{im_0t+1} \} \right] + \epsilon_{im_0t}. \quad (4)$$

Here, ϵ_{ijt} and ϵ_{im_0t} are assumed to follow the standard Type-I Extreme Value distribution and *i.i.d.* over time, products and consumer types.

Consumer Belief

Most of the literature on estimating dynamic demand models assumes that consumer purchase decisions are only based on a scalar state variable (the inclusive value) which follows an AR(1) process.⁸ Such a functional form restriction on consumer beliefs is difficult to reconcile with a supply model. This paper considers an alternative where consumers have rational expectations regarding the future environment. They can calculate the equilibrium strategies for all market participants as well as their own expected utility.

⁸Carranza (2007), Lee (2010), Gowrisankaran and Rysman (2006), Hendel and Nevo (2007).

Information Set

Let \mathbf{s}_{mt} denote the information set about submarket m at time t . It includes (1) the time period, t ; (2) the set of available products, J_{mt} ; (3) the observed and unobserved product characteristics of each available product, $\mathbf{X}_{mt} \equiv \{X_{jt}\}_{j \in J_{mt}}$ and $\xi_{mt} \equiv \{\xi_{jt}\}_{j \in J_{mt}}$; and (4) the mass of consumers remaining, $\mathbf{n}_{mt} \equiv \{n_{mit}\}_{i=1}^I$, where n_{mit} is the number of type- i -consumers who have not purchased any product in the submarket m at the beginning of period t . Besides, consumers also can observe the price of each available product, $\mathbf{p}_{mt} \equiv \{p_{jt}\}_{j \in J_{mt}}$, and their own demand shocks in submarket m , $\epsilon_{mit} = (\{\epsilon_{ijt}\}_{j \in J_{mt}}, \epsilon_{im0t})$.

Software Purchase

Let $G_i(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ denote the expected value function. Then, it can be written as

$$G_i(\mathbf{s}_{mt}, \mathbf{p}_{mt}) = \log\left\{ \sum_{j \in J_{mt}} \exp(X_{jt}\psi - \varphi_i p_{jt} + \xi_{jt}) + \exp[\beta_c EG_i(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt})] \right\}. \quad (5)$$

The probability that a type i consumer purchases the software product $j \in J_{mt}$ at time t is

$$b_{ij}(\mathbf{s}_{mt}, \mathbf{p}_{mt}) = \frac{\exp(X_{jt}\psi - \varphi_i p_{jt} + \xi_{jt})}{\exp[\beta_c EG_i(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt})] + \sum_{j \in J_{mt}} \exp(X_{jt}\psi - \varphi_i p_{jt} + \xi_{jt})}. \quad (6)$$

Hence, the demand for the software product $j \in J_{mt}$ at time t is $q_{jt} = \sum_i n_{mit} b_{ijt}$.

Consumer Distribution

In dynamic models of discrete choice demand, the mass of consumers remaining in a submarket is endogenous to the historic entry and pricing behavior of all firms in that submarket. So, the dynamics of entry and pricing introduce a dynamic evolution of the consumer distribution in the software submarket m as follows:

$$n_{mit+1} = n_{mit} \left(1 - \sum_{j \in J_{mt}} b_{ijt}\right) + q_{ilt}^h \quad (7)$$

where $n_{mit} \left(1 - \sum_{j \in J_{mt}} b_{ijt}\right)$ is the mass of consumers not buying in period t and remaining for the next period; and q_{ilt}^h is the mass of new consumers who purchase the compatible hardware, as described

in the previous subsection.

Total Software Utility

In the previous subsection, I specify that the consumption value of a hardware depends on the total utility from being able to purchase its affiliated software, Γ_{ilt} . To close the demand side of the model, I need to link it to the value of being on the software market, given by

$$\Gamma_{ilt} = \sum_m G_i(\mathbf{s}_{mt}, \mathbf{p}_{mt})$$

3.3 Software Pricing and Entry

I now describe how software firms behave in a submarket m , that is, how the incumbents set their optimal sequence of prices over time and how potential entrants make their optimal choices of whether or not to release a new product.

3.3.1 A Firm's Problem

Incumbent Firms. Let c be the marginal cost of production which is assumed to be constant over time. An incumbent firm's one-period profit depends on its own price choice this period (p_{jt}) but also on its competitors' prices (\mathbf{p}_{-jt}); moreover, it also depends on the state vector \mathbf{s}_{mt} in the submarket m which includes the set of available products, product characteristics and the consumer distribution. An incumbent's optimization problem is to pick a price to maximize its own discounted profit,

$$\begin{aligned} \Pi_{jt}(\mathbf{s}_{mt}, p_{jt}, \mathbf{p}_{-jt}) &= \pi_{jt}(\mathbf{s}_{mt}, p_{jt}, \mathbf{p}_{-jt}) \\ &+ E \left\{ \sum_{\tau=t+1}^T \beta_f^{\tau-t} \left[\max_{p_{j\tau}} \pi_{j\tau}(\mathbf{s}_{m\tau}, p_{j\tau}, \mathbf{p}_{-j\tau}) \right] \mid \mathbf{s}_{mt}, \mathbf{p}_{mt} \right\}, \end{aligned}$$

where β_f is the firm's discount factor and $\pi_{jt}(\mathbf{s}_{mt}, p_{jt}, \mathbf{p}_{-jt}) = (p_{jt} - c)q_j(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ is the one-period profit.

Potential Entrants. Every period, there is finite number of potential entrants outside the submarket m . Let E_{mt} denote the set of potential entrants. Each potential entrant $j \in E_{mt}$ first draws an entry cost from a known distribution and then decides whether to enter the submarket m . Potential entrants are short lived and base their entry decisions on the net present value of

entering today; they do not take the option value of delaying entry into account. If it enters, it pays the entry cost and starts to earn profit next period; if not, it earns zero profits. The entry cost is assumed to be $\gamma + \nu_{jt}$ where γ is the component that is common to all firms in the submarket and ν_{jt} is a private information shock which is assumed to be independently and identically distributed across firms and periods with c.d.f. $F_\nu(\cdot)$. Let $y_{jt+1} = 1$ iff entrant j enters. Essentially, a potential entrant j 's optimization problem is to compare the entry cost and the expected profit.

3.3.2 Perceived Strategy Function

Because the entry decision of a potential entrant depends on its own entry cost shock ν_{jt} which is unobservable to consumers and other firms, other agents cannot know exactly a potential entrant's entry strategy even if they can observe the actual outcomes. We can define a set of conditional choice probabilities for $j \in E_{mt}$ such that

$$\rho_j(\mathbf{s}_{mt}) = \int I(y_j(\mathbf{s}_{mt}, \nu_{jt}) = 1) dF_\nu(\nu_{jt}), \quad (8)$$

where $I(\cdot)$ is the indicator function. The probabilities represent the expected behavior of entrant j from the point of view of consumers and the rest of the firms. The game has a Markov structure, and I assume that each firm plays Markov strategies. That is, if $\mathbf{s}_{mt} = \mathbf{s}_{m't}$, then firm j 's decision in submarket m and m' are the same. Let $\Psi = \{\Psi_j(\mathbf{s}_{mt})\}$ be a set of strategy functions or decision rules, one for each firm, with $\Psi_j(\mathbf{s}_{mt}) = p_j(\mathbf{s}_{mt})$ if j is an incumbent firm and $\Psi_j(\mathbf{s}_{mt}) = \rho_j(\mathbf{s}_{mt})$ if j is a potential entrant.

3.3.3 Incumbent's Bellman Equation

Let $V_j(\mathbf{s}_{mt} | \Psi)$ denote the expected net present value of all future cash flows to incumbent firm $j \in J_{mt}$ at state vector \mathbf{s}_{mt} , computed under the presumption that consumers respond optimally and other firms follow their strategies in Ψ . By Bellman's principle of optimality, it can be written as

$$V_j(\mathbf{s}_{mt} | \Psi) = \max_{\tilde{p}_{jt}} \pi_{jt}(\mathbf{s}_{mt}, \tilde{p}_{jt}, p_{-jt}) + \beta_f E[V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, \tilde{p}_{jt}, \Psi_{-j}], \quad (9)$$

where

$$E[V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, p_{jt}, \Psi_{-j}] = \int_{\xi_{mt+1}} \left[\sum_{y_{mt+1}} V_j(\mathbf{s}_{mt+1} | \Psi) f_j(y_{mt+1} | \mathbf{s}_{mt}, p_{jt}, \Psi_{-j}) \right] d\xi_{mt+1}$$

is the expected value function conditional on firm j choosing p_{jt} and the other firms behaving according to Ψ . Here, the conditional transition probability function is given by

$$f_j(y_{mt+1} | \mathbf{s}_{mt}, p_{jt}, \Psi_{-j}) = \prod_{l \in E_{mt}} \rho_l(\mathbf{s}_{mt})^{y_{lt+1}} (1 - \rho_l(\mathbf{s}_{mt}))^{1-y_{lt+1}}.$$

The optimal pricing strategy in response to profile Ψ is the solution of the right hand side of equation (9), denoted as $p_j(\mathbf{s}_{mt} | \Psi)$.

3.3.4 Entrant's Bellman Equation

Let $V_j^e(\mathbf{s}_{mt}, \nu_{jt} | \Psi)$ denote the expected net present value of all future cash flows to potential entrant $j \in E_{mt}$ at state vector \mathbf{s}_{mt} and entry cost shock ν_{jt} , computed under the presumption that consumers respond optimally and other firms behave according to strategy profile Ψ . It is defined recursively by the solution

$$V_t^e(\mathbf{s}_{mt}, \nu_{jt} | \Psi) = \max_{y_{jt+1}} \tilde{y}_{jt+1} \{-\gamma - \nu_{jt} + \beta_f E[V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, \Psi]\},$$

where

$$E[V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, \Psi] = \int_{\xi_{mt+1}} \left[\sum_{y_{mt+1}} V_j(\mathbf{s}_{mt+1} | \Psi) f_j(y_{mt+1} | \mathbf{s}_{mt}, \Psi) \right] d\xi_{mt+1}$$

is the expected value function conditional on on firm j choosing entering and the other firms behaving according to strategy profile Ψ . Here, the conditional transition probability function is given by

$$f_j(y_{mt+1} | \mathbf{s}_{mt}, \Psi) = \prod_{l \in E_{mt}, l \neq j} \rho_l(\mathbf{s}_{mt})^{y_{lt+1}} (1 - \rho_l(\mathbf{s}_{mt}))^{1-y_{lt+1}}$$

The optimal entry decision follows a cutoff rule characterized by

$$y_j(\mathbf{s}_{mt}, \nu_{jt} | \Psi) = \begin{cases} 1, & \text{if } \nu_{jt} \leq \bar{\nu}_j(\mathbf{s}_{mt} | \Psi) \\ 0, & \text{otherwise} \end{cases}$$

where

$$\bar{\nu}_j(\mathbf{s}_{mt} | \Psi) = \beta_f E[V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, \Psi] - \gamma$$

is the cutoff entry cost shock for which the potential entrant is indifferent between entering and staying out of the submarket. Then, the probability of entering is

$$\rho_j(\mathbf{s}_{mt} | \Psi) = \int I[\nu_{jt} \leq \bar{\nu}_j(\mathbf{s}_{mt} | \Psi)] dF_\nu(\nu_{jt}) = F_\nu[\bar{\nu}_j(\mathbf{s}_{mt} | \Psi)].$$

Therefore, the unconditional Bellman equation of a potential entrant j can be written as

$$V_j^e(\mathbf{s}_{mt} | \Psi) = \max_{\tilde{\rho}_{jt+1}} - \int_{\nu_{jt} < F_\nu^{-1}(\tilde{\rho}_{jt+1})} \nu_{jt} dF_\nu(\nu_{jt}) + \tilde{\rho}_{jt+1} \{-\gamma + \beta_f E[V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, \Psi]\}. \quad (10)$$

3.4 Equilibrium Concept

In this paper, a hardware's value depends on the total utility from affiliated software, and thus any software firm's actions should affect hardware adoption. However, to simplify the model, I assume that software firms do not take that effect into account when they make their decisions. One possible condition is that they are very small compared to the whole market. Under this assumption, strategic interactions only occur between software firms and consumers who have already joined a hardware platform.

An action or decision for consumer i in state (\mathbf{s}, \mathbf{p}) specifies the probability vector that the consumer purchases each software product available: $\mathbf{b}_i(\mathbf{s}, \mathbf{p}) \in \mathbb{C}$, where $\mathbb{C}(\mathbf{s}, \mathbf{p}) = [0, 1]^{\#J}$ denote a consumer's feasible actions. An action or decision for firm j in state \mathbf{s} specifies either the price that the incumbent charges or the probability that the entrant enters: $\Psi_j(\mathbf{s}) \in \mathcal{F}_j(\mathbf{s})$, where $\mathcal{F}_j(\mathbf{s})$ denotes the space of firm j 's feasible actions in state \mathbf{s} with $\mathcal{F}_j(\mathbf{s}) = [0, \bar{p}]$ if firm j is an incumbent and $\mathcal{F}_j(\mathbf{s}) = [0, 1]$ if firm j is a potential entrant.

A strategy or policy for a consumer i or a firm j is a single function from the state space into the action space. Such a strategy is called Markovian because it is restricted to be a function of the current state rather than the entire history of the game.

A Markov Perfect equilibrium (MPE) in this game is a set of policy functions including each consumer's probability of purchasing and each firm's strategy, $\{\mathbf{b}^*(\mathbf{s}, \mathbf{p}), \Psi^*(\mathbf{s})\}$ and a set of value functions $\{G^*(\mathbf{s}, \mathbf{p}), V^*(\mathbf{s}), V^{e*}(\mathbf{s})\}$ such that :

Consumers: At any state vector (\mathbf{s}, \mathbf{p}) , given the strategies of all firms $\Psi^*(\mathbf{s}, \mathbf{p})$, each consumer's expected value function $G_i^*(\mathbf{s}, \mathbf{p})$ satisfies equation (5) and his/her probability of purchasing $b_{ij}^*(\mathbf{s}, \mathbf{p})$

follows the equation (6).

Software Firms: At any state vector \mathbf{s} , given other firms' strategy $\Psi_{-j}^*(\mathbf{s})$, for any incumbent firm j , its expected value function $V_j^*(\mathbf{s})$ satisfies the incumbent's Bellman equation (9) and its best strategy $p_j^*(\mathbf{s})$ solves the right hand side of that Bellman equation (9); and for any entrant firm j , its expected value function $V_j^{e*}(\mathbf{s})$ satisfies the entrant's Bellman equation (10) and its best strategy $\rho_j^*(\mathbf{s})$ solves the right hand side of that Bellman equation (10).

4 Bayesian Estimation

In this section, I describe the proposed modified Bayesian MCMC estimation method in detail. Let θ denote the vector of parameters in the model that need to be estimated. Let *data* denote all the data available for estimation which includes two parts: (i) the characteristics, prices, and quantity sold of each hardware product in each time period; and (ii) the availability, characteristics, prices, and quantity sold of each software product in each time period across M independent markets. Hence, $data = \{\mathbf{X}_t^h, \mathbf{p}_t^h, \mathbf{q}_t^h, \{\mathbf{y}_{mt}, \mathbf{X}_{mt}, \mathbf{p}_{mt}, \mathbf{q}_{mt}\}_{m=1}^M\}_{t=1}^{T_d}$, where T_d is the number of time periods in the data set. I assume that the data are generated from the model presented in the previous section.

4.1 Posterior with Multiple Equilibria

One issue associated with estimating dynamic games is the multiple equilibria problem. The standard assumption in the empirical literature is to assume that only one equilibrium is being played in the data. Ignoring the presence of multiple equilibria can lead to biases in the estimation of the parameters and in the policy recommendations of the results. To accommodate the multiple equilibria in complete-information static games, Bajari, Hong and Ryan (2010) suggest and estimate an equilibrium selection mechanism. In the spirit of their method, I also adopt a stochastic equilibrium selection mechanism to deal with the multiple equilibria problem.

I assume that there exist at most R candidate equilibria.⁹ Let $\Psi_r^*(\mathbf{s}, \theta)$ denote the r th candidate equilibrium supported by the parameter θ and the state \mathbf{s} . Assume that each submarket is randomly assigned a candidate equilibrium with some per-determined probability. In essence, each

⁹The number of equilibrium is potentially endogenous and in general one can not make an assumption on an endogenous outcome. Here, I just specify the maximum of the number of equilibria. If one candidate equilibrium is not an equilibrium, then the probability associated with that candidate equilibrium would be zero.

submarket is assumed to play only one equilibrium. From the researcher’s point of view, this can be interpreted as a stochastic equilibrium selection mechanism. Let λ_r denote the probability of the r th candidate equilibrium is being assigned, with $\sum_{r=1}^R \lambda_r = 1$. Furthermore, let $\mathcal{L}(data \mid \Psi_r^*, \theta)$ denote the likelihood conditional on that all agents play the r th candidate equilibrium Ψ_r^* . Then, the unconditional likelihood is the expectation of the conditional likelihood,

$$\mathcal{L}(data \mid \theta) = \sum_{r=1}^R \lambda_r \mathcal{L}(data \mid \Psi_r^*, \theta). \quad (11)$$

Rather than using a nested-fixed point or two-step method to maximize the likelihood above, I employ the Bayesian MCMC method to sample the parameter vector θ and the equilibrium probability vector λ jointly from the posterior,

$$\mathcal{P}(\theta, \lambda \mid data) \propto \mathcal{L}(data \mid \theta) \pi_\lambda(\lambda) \pi_\theta(\theta), \quad (12)$$

where $\pi_\theta(\cdot)$ is the prior distribution of the parameter vector θ and $\pi_\lambda(\cdot)$ is the prior distribution of the equilibrium probability vector λ . In essence, the equilibrium probabilities are treated as parameters to be estimated along with the structural parameters. Generally speaking, it is difficult to obtain proofs of existence of a Markov-perfect equilibrium in pure strategies for dynamic oligopoly models. The difficulty is increased by the absence of easily verifiable conditions for uniqueness of the equilibrium in a subgame starting from a particular state. One way is to impose additional restrictions on the model to ensure that each agent’s best reply is always unique.¹⁰ Even though this paper does not explore much on this issue, it is definitely an important research question that needs further exploration.

4.2 Likelihood Contributions

In this subsection, I specify the likelihood conditional on the r th candidate equilibrium in the equation (11), $\mathcal{L}(data \mid \Psi_r^*, \theta)$.

The demand for hardware is a dynamic discrete choice model. I assume that the unobserved (to researcher) platform-specific demand shifters ξ_{jt}^h are normally distributed with mean zero and

¹⁰Doraszelski and Satterthwaite (2010) provide a condition on model primitives that guarantees the existence of pure strategy equilibrium in Ericson-Pakes-style models. To guarantee the uniqueness of the entry and exit best responses, they assume that the densities of the scrap values and setup values are continuous. Furthermore, they prove that if the transition function is unique investment choice (UIC) admissible, then a firm’s investment decision is indeed uniquely determined.

variance $\sigma_{\xi_h}^2$, independent across all products and over time. The distribution of the aggregate demand shocks generate the distribution of the units sold of each existing product in each time period. Conditional on the state \mathbf{S}_t and all firms playing the candidate equilibrium Ψ_r^* , the joint density of the sales of all hardware at time t is

$$\mathcal{L}_{dh}(\mathbf{q}_t^h \mid \mathbf{S}_t, \Psi_r^*; \theta) = \prod_l \left[\phi(\xi_{lt}^h / \sigma_{\xi_h}) / \sigma_{\xi_h} \right] \mid \left(J_{(\mathbf{q}_t^h \rightarrow \xi_{lt}^h)} \right)^{-1} \mid, \quad (13)$$

where $\phi(\cdot)$ is the *pdf* of the standard normal distribution and $J_{(\mathbf{q}_t^h \rightarrow \xi_{lt}^h)}$ is the Jacobian matrix.

The demand for software is also a dynamic discrete choice model. To specify the likelihood contribution of the demand side, I add one additional assumption that the unobserved (to researchers) game-specific demand shifters ξ_{jt} are normally distributed with mean zero and variance σ_{ξ}^2 , independent across all products and over time.¹¹ The distribution of the aggregate demand shocks generate the distribution of the units sold of each existing product in each time period. Conditional on the state $(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ and all firms playing the candidate equilibrium Ψ_r^* , the joint density of the sales of all existing products in submarket m at time t is

$$\mathcal{L}_{dg}(\mathbf{q}_{mt} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt}, \Psi_r^*; \theta) = \prod_{j \in J_{mt}} \left[\phi(\xi_{jt} / \sigma_{\xi}) / \sigma_{\xi} \right] \mid \left(J_{(\mathbf{q}_{mt} \rightarrow \xi_{mt})} \right)^{-1} \mid,$$

To evaluate the likelihood, we need to derive ξ_{jt} , which is described in section 4.4, and evaluate the Jacobian, $J_{(\mathbf{q}_{mt} \rightarrow \xi_{mt})}$, which is derived in Appendix C.

Therefore, the likelihood contribution of the demand side is

$$\mathcal{L}_d(\mathbf{q}_t^h, \{\mathbf{q}_{mt}\}_{m=1}^M \mid \mathbf{S}_t, \Psi_r^*; \theta) = \mathcal{L}_{dh}(\mathbf{q}_t^h \mid \mathbf{S}_t, \Psi_r^*; \theta) \prod_m \mathcal{L}_{dg}(\mathbf{q}_{mt} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt}, \Psi_r^*; \theta) \quad (14)$$

Next I specify the likelihood contribution of the software pricing policy function. Let \tilde{p}_{jt} and p_{jt}^* denote the observed price and the actual price of product j at time t , respectively. Assume that the observed price is proportional to the actual price, that is, $\tilde{p}_{jt} = p_{jt}^* \varsigma_{jt}$, where ς_{jt} is the measurement error that reflects discrepancies between the observed prices and the actual prices.¹² Furthermore,

¹¹In the context of sports video games, ξ_{jt} may capture such demand shocks as events related to the celebrities on whom game characters are based, e.g., their performance in major tournaments and even their scandals. Those shocks occur independently across products and over time and thus it is reasonable to assume no cross-correlation and no auto-correlation.

¹²In the dataset, I can observe the revenue (measured in dollars) and the units sold in each month of each game title released during the sample period. The price in each month is measured by the average price in that month, i.e., the ratio of the revenue over the units sold. However, this measurement of price contains some measurement error because the actual price changes during each month. Hence, I add the measurement error term ς_{jt} .

it is assumed to follow a log-normal distribution with mean zero and variance σ_ζ^2 , independent over time and across products. Hence, conditional on the state vector \mathbf{s}_{mt} and an equilibrium profile Ψ_r^* , the likelihood contribution of incumbent $j \in J_{mt}$ at time t is given by

$$\mathcal{L}_p(p_{jt} \mid \mathbf{s}_{mt}, \Psi_r^*; \theta) = \frac{1}{\sigma_\zeta} \phi \left(\frac{\ln[\tilde{p}_{jt}/p_j(\mathbf{s}_{mt}, \theta \mid \Psi_r^*)]}{\sigma_\zeta} \right). \quad (15)$$

To specify the likelihood contribution of the software entry policy function, I assume that the entry cost shocks follow an independent normal distribution with mean zero and variance σ_ν^2 .¹³ Hence, conditional on the state vector \mathbf{s}_{mt} and the equilibrium profile Ψ_r^* , the likelihood contribution of entrant $j \in E_{mt}$ is

$$\begin{aligned} \mathcal{L}_y(y_{jt+1} \mid \mathbf{s}_{mt}, \Psi_r^*; \theta) &= \left(\Phi \left[\frac{\beta_f E[V_j(\mathbf{s}_{mt+1} \mid \mathbf{s}_{mt}, \Psi_r^*; \theta)] - \gamma}{\sigma_\nu} \right] \right)^{y_{jt+1}} \\ &\times \left(1 - \Phi \left[\frac{\beta_f E[V_j(\mathbf{s}_{mt+1} \mid \mathbf{s}_{mt}, \Psi_r^*; \theta)] - \gamma}{\sigma_\nu} \right] \right)^{1-y_{jt+1}}. \end{aligned} \quad (16)$$

Therefore, the likelihood conditional on all software firms playing the candidate equilibrium Ψ_r^* can be written as

$$\begin{aligned} \mathcal{L}(\text{data} \mid \Psi_r^*, \theta) &= \prod_{t=1}^{T_d} \mathcal{L}_d(\mathbf{q}_t^h, \{\mathbf{q}_{mt}\}_m \mid \mathbf{S}_t, \Psi_r^*; \theta) \\ &\times \prod_{m=1}^M \left\{ \prod_{j \in J_{mt}} \mathcal{L}_p(p_{jt} \mid \mathbf{s}_{mt}, \Psi_r^*; \theta) \prod_{j \in E_{mt}} \mathcal{L}_y(y_{jt+1} \mid \mathbf{s}_{mt}, \Psi_r^*; \theta) \right\}. \end{aligned} \quad (17)$$

The computation of the likelihood above is discussed in detail in section 4.4.

4.3 Modified Metropolis-Hastings Algorithm

The posterior distribution in equation (12) is a high-dimensional and complex function of the parameters. It is known that, instead of drawing the entire parameter vector at once, it is often simpler to partition it into blocks and draw the parameters of each block separately given the other parameters. Based on the model, I partition all parameters into three blocks: θ_1 includes all parameters of the demand side, i.e., all the parameters in the utility functions; θ_2 includes all parameters directly affecting incumbent firms' pricing decisions, i.e., unit cost of selling software products on each platform and the variance of pricing error; and θ_3 includes all parameters directly affecting

¹³We should notice that this assumption on entry cost shocks may not hold if we consider learning-by-doing or technology spillover effect.

entrants' entry decisions, i.e., mean of entry cost for each platform and the variance of entry cost shocks. Together with the candidate equilibrium probability vector, λ , we have four blocks.

Consider a particular iteration k . For each block l , $l = 1, \dots, 4$, the procedure goes as follows:

The first step is to draw the candidate parameter vector $\theta_l^{*(k)}$ from a proposal density. As usual in the literature,¹⁴ I use the Random-Walk (RW) Metropolis chain as the proposal density

$$\theta_l^{*(k)} = \theta_l^{(k-1)} + MVN(0, \kappa \Sigma_l)$$

where Σ_l is the candidate covariance matrix and κ is a scaling constant.

The second step is to construct the acceptance-rejection ratio. For the first three blocks, the ratio is given by

$$\eta_l^{(k)*} = \frac{\left[\sum_{r=1}^R \lambda_r^{(k-1)} \mathcal{L}_l(\cdot \mid \Psi_r^*, \theta_l^{(k)*}, \theta_{-l}^{(k-1)}) \right] f_l(\theta_l^{(k)*} \mid \theta_l^{(k-1)}) \pi_l(\theta_l^{(k)*})}{\left[\sum_{r=1}^R \lambda_r^{(k-1)} \mathcal{L}_l(\cdot \mid \Psi_r^*, \theta_l^{(k-1)}, \theta_{-l}^{(k-1)}) \right] f_l(\theta_l^{(k-1)} \mid \theta_l^{(k)*}) \pi_l(\theta_l^{(k-1)})},$$

where $\mathcal{L}_l(\cdot \mid \Psi_r^*, \theta)$ equals to equation (14), (15) and (16) for $l = 1, 2, 3$, respectively; $f_l(\theta_l^{(k)*} \mid \theta_l^{(k-1)})$ is the transition probability, and $\pi_l(\theta_l^{(k)*})$ is the prior distribution. For the last block, the ratio is

$$\eta_4^{(k)*} = \frac{\left[\sum_{r=1}^R \lambda_r^{(k)*} \mathcal{L}(\text{data} \mid \Psi_r^*, \theta^{(k)}) \right] f_\lambda(\lambda^{(k)*} \mid \lambda^{(k-1)}) \pi_\lambda(\lambda^{(k)*})}{\left[\sum_{r=1}^R \lambda_r^{(k-1)} \mathcal{L}(\text{data} \mid \Psi_r^*, \theta^{(k)}) \right] f_\lambda(\lambda^{(k-1)} \mid \lambda^{(k)*}) \pi_\lambda(\lambda^{(k-1)})},$$

where $\mathcal{L}(\text{data} \mid \Psi_r^*, \theta)$ is given by the equation (17).

Lastly, we accept the candidate parameter vector $\theta_l^{*(k)}$ with probability $\min\{\eta_l^{(k)*}, 1\}$.

4.4 Computation of Likelihoods

The computation difficulty of evaluating the likelihood in equation (17) comes in two parts. One part is computing each agent's expectation of the next-period value function given the state variable, its own action and other agents' actions in this period. The other part is computing the equilibrium strategies of all agents in the game given the state variable. In this subsection, I present my method to address those challenges in detail.

¹⁴See Jiang, Manchanda and Rossi (2009), and Imai, Jain, and Ching (2009).

4.4.1 Computing the Expected Value Functions

The first challenge in estimating dynamic games is to compute the expected value function at a particular state point. The conventional algorithm iterates the Bellman operator until convergence. It is computationally difficult for relatively complicated models. The IJC algorithm proposes a non-parametric kernel approach to approximate the expected value function using the weighted average of pseudo-value functions of most recent iterations. Unlike conventional approaches, in which value functions need to be computed at all or a subset of pre-determined grid points in all periods (e.g., Rust 1997), the new algorithm computes pseudo-value functions at only one randomly drawn state point in each period, and the integration of the continuation value with respect to continuous state variables can simply be done by the weighted average of past pseudo-value functions. Thus, it has the potential to reduce the computational burden.

One issue in applying the IJC algorithm to the current model is that it is a finite-period model which is non-stationary; however, the original IJC algorithm applies to stationary dynamic programming problems. Hence, I compute and store the pseudo-value functions for each period, and approximate the expected value functions in period t using the set of pseudo-value functions in period $t + 1$.

For consumers, the expectation of the next-period value function at next-period state $(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1})$ in iteration k is approximated as

$$\begin{aligned} \hat{E} \left[G_i^{(k)}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1}, \theta \mid \Psi_r^*) \right] &= \sum_{n=1}^{N(k)} G_i^{(k-n)}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1}, \theta^{*(k-n)} \mid \Psi_r^*) \\ &\quad \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})}, \end{aligned} \quad (18)$$

where $K_h(\cdot)$ is a multivariate kernel with bandwidth $h > 0$, and $G_i^{(k)}(\mathbf{s}_{mt}, \mathbf{p}_{mt}, \theta \mid \Psi_r^*)$ is the consumer's pseudo-value function at state $(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ conditional on all firms playing the candidate equilibrium Ψ_r^* :

$$\begin{aligned} G_i^{(k)}(\mathbf{s}_{mt}, \mathbf{p}_{mt}, \theta \mid \Psi_r^*) &= \log \left\{ \sum_{j \in J_{mt}} \exp(x_{jt} \varphi^x - \varphi_i^p p_{jt} + \xi_{jt}) + \exp[\beta_c] \right. \\ &\quad \left. \times \sum_{\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1}} \hat{E} G_i^{(k)}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1}, \theta \mid \Psi_r^*) f(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt}, \Psi_r^*) \right\}. \end{aligned}$$

The approximated expected value function given by equation (18) is the weighted average of the pseudo-value functions of $N(k)$ most recent iterations. IJC (2009) show that, as the MCMC iterations and the number of past iterations for approximating the expected value functions increase,

the pseudo-value function converges to the true value functions, and the posterior parameter draws based on the pseudo-value functions will also converge to the true posterior distributions. Moreover, the convergence of the approximated expected value function to the true value function requires that $N(k) \rightarrow \infty$ and $k - N(k) \rightarrow \infty$ as $k \rightarrow \infty$. The IJC algorithm is similar to the idea of Pakes and McGuire (2001) in which the expected value function is approximated by the average of the value functions of the past N iterations that have the same state space point as the current state space point. In the IJC algorithm, the averages are taken over the value functions that have the same state as the current state as well as parameters that are close to the current parameter.

A similar method applies to computing the expectation of incumbent's next-period value function and the expectation of entrant's next-period value function:

$$\hat{E} \left[V_j^{(k)}(\mathbf{s}_{mt+1}, \theta \mid \Psi_r^*) \right] = \sum_{n=1}^{N(k)} V_j^{(k-n)}(\mathbf{s}_{mt+1}, \theta^{*(k-n)} \mid \Psi_r^*) \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})} \quad (19)$$

where $V_j^{(k)}(\mathbf{s}_{mt}, \theta \mid \Psi_r^*)$ is the incumbent firms' pseudo-value functions at state \mathbf{s}_{mt} conditional on that all other firms playing the candidate equilibrium Ψ_r^* :

$$V_j^{(k)}(\mathbf{s}_{mt} \mid \Psi_r^*) = \max_{\tilde{p}_{jt}} \pi_{jt}(\mathbf{s}_{mt}, \tilde{p}_{jt}, p_{-jt}) + \beta_f \int \hat{E} V_j^{(k)}(\mathbf{s}_{mt+1}, \theta \mid \Psi_r^*) f_j(\mathbf{s}_{mt+1} \mid \mathbf{s}_{mt}, \tilde{p}_{jt}, \Psi_r) d\mathbf{s}_{mt+1}.$$

To obtain the expected value functions, however, we still need to compute equation (18) and (19) for every possible state space point. Due to the ‘‘curse of dimensionality’’,¹⁵ it is computationally burdensome and even impossible to achieve it even with the approximation method proposed above. In next subsection, I will describe my strategy to solve this large space problem.

4.4.2 State Space Problem

In the literature, the simulation and interpolation approach proposed by Keane and Wolpin (1994) has been the most widely used for applications with finite horizon problems with large state spaces. This method obtains simulated-based approximations to the expected value function only at a (randomly chosen) subset of the state points every period, and obtains the expected values at other points as the predicted values from a regression function which is estimated from the points in that subset.

In the spirit of Keane and Wolpin's method, I propose a new algorithm to deal with the large

¹⁵The number of possible state vectors grows geometrically in the number of agents and exponentially in the number of states per agent. For example, if we have N agents, K state variables each taking on M distinct values, then the number of possible state vectors for each agent is $(KM)^N$.

state space problem. In the first step, I randomly choose a subset of the state points every period, and obtain the expected values at those points with the non-parametric approximation approach described in the previous subsection. Next, I interpolate the value functions with a quadratic-in-states polynomial approximation in that subset. Lastly, for each current state, I simulate a next-period-state using the equilibrium strategies, and then use the predicted value at that simulated next-period-state as the continuation value. In practice, I simulate the next-period-state for finite times and then take the average of the predicted values. This algorithm is similar to Pakes and McGuire (2001) where they never attempt to obtain accurate policies on the entire state space, just on a recurrent class of points.

This algorithm significantly alleviates the computational burden and makes it possible to estimate models with very large state spaces and rich structure. However, we also should notice that estimators of structural parameters are not consistent as long as interpolation is used, because the approximation errors in the expected value functions enter nonlinearly in optimization problems.¹⁶

Recall that the state vector in the model includes the availability and characteristics of each product in the submarket, and the distribution of remaining consumers of each type. Besides, consumers can also observe the price of each product in the submarket. Among those state variables, the product characteristics evolves exogenously and deterministically; the consumer distribution evolves deterministically depending on consumers purchase choices; the product availability depends on all potential entrants' entry choice up to the previous period; and the price is chosen by incumbent firms based on the state vector. In terms of computation, it is extremely difficult and even impossible to include all of those state variables. Hence, I characterize each agent's state vector as follows.

Consumers trace the time periods to the end; the number of products available for purchase; the distribution of remaining consumers of each type; his/her own mean utility from the top-ranked product; and the average of his/her mean utility from all existing products. Incumbent firms trace the time periods to the end; the number of competitors in the same submarket; the distribution of remaining consumers of each type; consumer's valuation of the top-ranked product; and consumer's valuation about its product. Potential entrants trace almost the same variables as incumbent firms do. The only difference is that they trace the expected value of new product instead of the value of its own product.

¹⁶Note that approximation error in the expected value function is not the only source of potential inconsistency, for example, discretization of continuous variables, approximate convergence of the Bellman operator in infinite horizon problems and others.

4.4.3 Computing the Pricing and Entry Functions

Another challenge in computing the likelihood is to compute the equilibrium of a dynamic game which is the fixed point of the best response system. In the literature, the nested fixed point approach computes the equilibrium numerically.¹⁷ However, applying it for relatively complicated models becomes extremely difficult and even impossible even for one guess of the parameter vector. The two-step approach (Bajari, Benkard and Levin, 2007), sidesteps the equilibrium computation step by substituting nonparametric functions of the data for the continuation values in the game, which is in general much computationally easier than the fixed point calculations. However, this approach suffers from a small sample bias problem and also can not deal easily with the unobservable state variable problem.

In this paper, I propose to use a kernel method to approximate the equilibrium strategies perceived by other agents using the pseudo-best response of the past iterations in which the parameter vector is “close” to the current parameter vector. The perceived equilibrium strategy of firm j in iteration k is computed as

$$\hat{\Psi}_{rj}^{(k)}(\mathbf{s}_{mt}, \theta) = \sum_{n=1}^{N(k)} \Psi_{rj}^{(k-n)}(\mathbf{s}_{mt}, \theta^{*(k-n)}) \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})}, \quad (20)$$

where $\Psi_{rj}^{(k)}$ is the pseudo-best response function in the iteration k . For incumbent firm j , the pseudo-best response in price is the solution to the incumbent firm’s optimization problem, and Appendix D presents the computation method in detail. For entrant j , the pseudo-best response in entering probability is the solution to the potential entrant’s optimization problem. Under the assumption of normally distributed entry cost shocks, it is

$$\rho_{rj}^{(k)}(\mathbf{s}_{mt}, \theta) = \Phi \left(\left[\beta_f \hat{E}V_{jt+1}^{(k)}(\cdot | \mathbf{s}_{mt}) - \gamma \right] / \sigma_v \right).$$

In essence, the equilibrium strategies perceived by other agents is approximated by the weighted average of pseudo-best response of the prior iterations. In terms of computation, this method is much easier than calculating the fixed point of the best response system. Moreover, similar to the idea of the IJC algorithm, as the MCMC iterations and the number of past iterations for approximating the perceived equilibrium strategies increase, the pseudo-best response function converges to the true best response function, and the posterior parameter draws based on the pseudo-best response

¹⁷The general idea is to start with an initial guess at the value function and substitute that into the RHS of the Bellman equation. Then, at each state point and for each agent, solve the maximization problem yielding a new estimate of the value function. Iterate this procedure until convergence. The literature of NFP approach includes Pakes and McGuire (1994), Pakes and McGuire (2001).

functions also converge to the true posterior distributions.¹⁸

4.4.4 Computing ξ_{lt}^h and ξ_{jt}

Once we obtain the consumer's expected value function, we can compute each consumer's probability of purchasing from equation (6) and then the predicted demand of each product. To obtain the likelihood contribution of demand in equation (13), I update the aggregate demand shocks based on the expression,

$$\xi_{jt}^{(k)} = \xi_{jt}^{(k-1)} + \ln(\tilde{q}_{jt}) - \ln\left(q_{jt}^{(k)}(\mathbf{s}_{mt}, \theta)\right),$$

where \tilde{q}_{jt} is the units sold observed in the data and $q_{jt}^{(k)}(\mathbf{s}_{mt}, \theta)$ is the predicted quantity using the demand shocks of the $(k-1)$ th iteration, $\xi_{mt}^{(k-1)}$. This procedure is similar to the inversion proposed by BLP (1995). The main difference is that, unlike BLP, consumers in my model maximize inter-temporal utility, implying that the corresponding aggregate demands, $q_{jt}(\mathbf{s}_{mt}, \theta)$, are a function of the consumer's value of waiting each period. Another difference is that, unlike BLP which iterates the aggregate demand shocks until convergence for any given parameter vector, I update it only once during each MCMC iteration. A similar procedure applies to computing the aggregate demand shocks of hardware, ξ_{lt}^h , given by

$$\xi_{lt}^{h(k)} = \xi_{lt}^{h(k-1)} + \ln(\tilde{q}_{lt}^h) - \ln\left(q_{lt}^{h(k)}(\mathbf{S}_t, \theta)\right).$$

5 Estimation Results

5.1 Econometric Details

Consumer Heterogeneity. For simplicity but without loss of generality, I assume two consumer types, high-type consumers and low-type consumers, with different sensitivity to price, φ_H and φ_L ($\varphi_H \leq \varphi_L$).¹⁹ Besides, it is necessary to choose an initial number of consumers, N_0 . Once this is pinned down, the future distribution of each consumer type is determined by the consumer purchase decisions of games and consoles. In particular, their purchase decision of games in a submarket determines the number of consumers remaining for the next period, and their purchase decision of consoles determines the number of new consumers who enter the game market next

¹⁸Here, I need to develop formal proofs. It is part of future research.

¹⁹The number of customer types (I) should be determined by adding types till one of the type sizes is not statistically different from zero (Besanko et al. 2003). Nair (2007) says that the estimates for the three-type model yielded several insignificant parameters and thus he presented the estimates for the two-type case.

period. In this paper, I choose N_0 to be 100 millions.

Number of Possible Equilibria. In this paper, I set the maximal number of possible equilibria to be 3. I determine this number by adding possible equilibria till it does not improve the fitting of the model.

Prior Setting. In order to estimate the model it is necessary to specify the prior distribution for the parameters and the equilibrium probabilities to be estimated. Consumer's preference to product characteristics (ψ, ψ_h) and consumer's sensitivity to price (φ_i) follow normal distribution with mean zero and large standard deviation. The initial share of high-type consumers (δ_H) follows a uniform distribution on the interval $[0,1]$. To guarantee that cost parameters and standard deviations are non-negative, their prior are log-normal distribution with mean zero and large standard deviation. Besides, all equilibrium probabilities λ_r follow uniform distribution on the interval $[0,1]$.

Initial Guess of Equilibrium Strategies and Value Functions. The initial guess of consumer value functions and incumbent value function are computed by assuming that both consumers and firms are myopic. The initial guess of product price is the predicted values from a hedonic regression of price on state variables. The initial guess of the entry probability is computed based on the initial value functions.

Discount Factors. Previous literature has noted that it is difficult to estimate discounted factors. So, I do not attempt to estimate the discount factors for consumers and firms (β_c and β_f). Instead, I set the discount rates to be 0.95 which is lower than the monthly interest rate. However, previous studies in experimental/behavior economics have found that the discount factor is lower than the interest rate, for example, Frederick et al. (2002). Moreover, it is relatively computationally burdensome to solve consumer and firm's inter-temporal problem at larger discount values.

5.2 Estimates

I draw 10,000 samples from the posterior distribution. It runs around 10 hours. But, I only use the last 5,000 samples to derive the posterior means, standard deviations and median. They are reported in table 2. Besides, I also compute those statistics from the last 2,500 samples. Since they are not statistically different, I conclude that the samples that I use to compute the posterior statistics are drawn from a stable distribution.

Table 2: Posterior Means and Standard Deviations

	Last 5,000 Samples		Last 2,500 Samples	
	Mean	Std dev	Mean	Std dev
Block 1: Demand for Software and Hardware				
ψ_1 (online rating score)	0.2015	0.0017	0.2020	0.0013
ψ_2 (game age if new)	-0.2892	0.0035	-0.2878	0.0034
ψ_3 (game age if old)	-0.1851	0.0039	-0.1879	0.0012
ψ_4 (Nov. dummy)	0.7466	0.0911	0.8243	0.0524
ψ_5 (Dec. dummy)	1.4790	0.0136	1.4804	0.0111
φ_H (H-type price sensitivity)	0.0179	0.0002	0.0178	0.0001
φ_L (L-type price sensitivity)	0.0643	0.0012	0.0651	0.0011
σ_ξ (std of software demand shocks)	1.9719	0.0263	1.9554	0.0152
ψ_{PS} (dummy for PS)	0.5016	0.0076	0.5054	0.0069
ψ_{N64} (dummy for N64)	1.4567	0.0186	1.4560	0.0148
$\sigma_{\xi h}$ (std of hardware demand shocks)	0.0972	0.0009	0.0969	0.0011
δ_H (initial share of H-type)	0.1542	0.0113	0.1571	0.0130
Block 2: Software Pricing				
c_1 (unit cost of games for Saturn)	13.5071	0.4893	13.7885	0.3948
c_2 (unit cost of games for PS1)	11.7552	0.4087	11.7007	0.2226
c_3 (unit cost of games for N64)	17.4948	0.4951	17.2980	0.5330
σ_ζ (std of price error)	1.0284	0.0312	1.0048	0.0098
Block 3: Software Entry				
γ_1 (mean of entry cost to Saturn)	4.8610	0.1676	4.7274	0.1183
γ_2 (mean of entry cost to PS1)	3.6620	0.0928	3.7035	0.0713
γ_3 (mean of entry cost to N64)	5.0965	0.1707	5.2098	0.1518
σ_ν (std of entry cost shocks)	2.3926	0.0952	2.4625	0.0679
Block 4: Equilibrium Prob.				
λ_1 (prob. of equm #1)	0.2321	0.0689	0.2889	0.0502
λ_2 (prob. of equm #2)	0.1386	0.0242	0.1328	0.0281
λ_3 (prob. of equm #3)	0.6293	0.0607	0.5783	0.4467

The signs of the consumer utility function coefficients are consistent with our expectation. People favor the games with high online rating score. They dislike games which have been in the market for a long time, partly because most sports games are designed based on the latest tournaments. They obtain extra utility when it is either November or December, probably because they can spend more time on playing games and can also play games with their friends during holiday season.

The price sensitivity is 0.0179 for high-type consumers and 0.0643 for low-type consumers. They are positive just because they enter the utility function as a negative term. Low-type consumers are seen to be more price sensitive than high-type consumers. Besides, high-type consumers corresponds to roughly 15% of the potential market at the beginning of the console lifecycle.

The cost per unit is around \$13.5, \$11.8 and \$17.5 for games released on Sega Saturn, PlayStation One and Nintendo 64, respectively. As Coughlan (2001) reported, the production/packaging cost for 32-bit CD-ROM games is around \$1.5 per disc. Hence, the royalty fee charged by the three consoles are \$12, \$10 and \$16 per copy.

The average entry cost of Sega Saturn, PlayStation One and Nintendo 64 are 4.861 million dollars, 3.662 million dollars and 5.097 million dollars, respectively. The standard deviation of entry cost is 2.393 million dollars.

5.3 Fit of Model

To examine the fit of the equilibrium model, I treat the posterior means of the last 5,000 samples as the estimated values of the parameters of the model, and simulate the equilibrium of the model. I now compare the predicted values to those observed in the data. Figure 4 plots the predicted and the observed installed base of users of the three consoles during the sample period. Overall, the model fits the data very well. Saturn’s installed base grows initially and stops growing two years after its release. Both PlayStation and Nintendo 64 grow all the time, and PlayStation is the biggest platform in the market.

Figure 4: Predicted vs. Observed Installed Base of Users

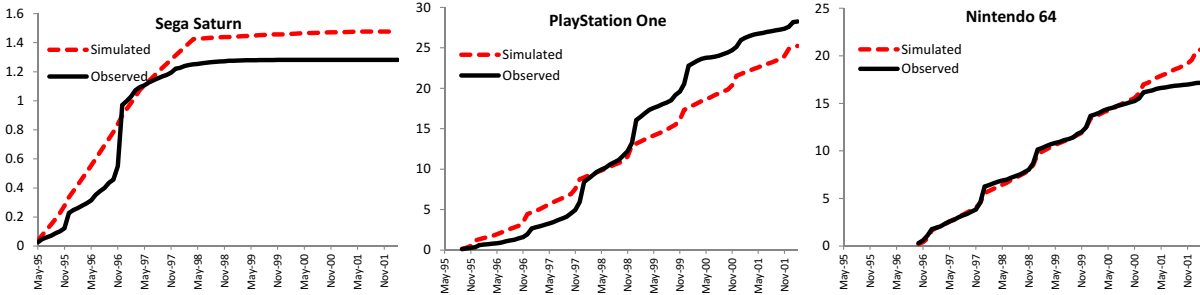
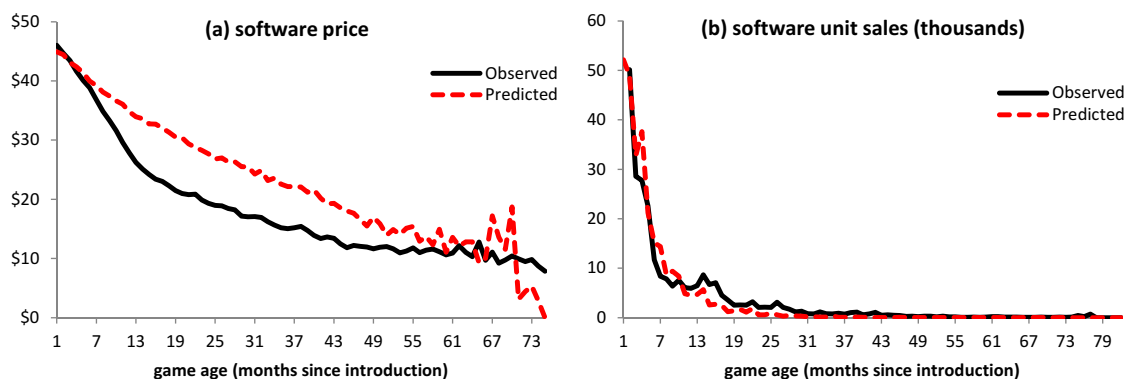


Figure 5 (a) plots the predicted and the observed software price across all sports games in the sample by game age (i.e., months since introduction). It indicates that the proposed model is able to explain the declining pattern of software price. However, the predicted price drops not so quickly as the observed price in the data. One possible reason is that the model does not consider the second-hand market which contributes to the declining price in the data. Besides, the predicted price turns to be gear-like when a game becomes really “old”. One possible reason is that only a few games can survive in the market for so long time, and the average price of very old games is more likely driven by some outliers. 5 (b) plots the predicted and the observed unit sales across

all sports games by game age. It shows that the model fits the data very well.

Figure 5: Predicted vs. Observed Software Price and Unit Sales in Age



5.4 Software Pricing

This subsection investigates the determinants of the prices of competing durable goods. When a software producer decides on its price, it takes into account the competition level in its submarket, the size of potential buyers of each type, and consumer forward-looking behavior.

To measure the importance of each effect, I approximate the pricing policy function as a polynomial of observed and unobserved state variables of the equilibrium model. In the model, the state space is huge and thus I include these variables: the number of active software products in the same submarket, number of active high-type consumers, number of active low-type consumers, the waiting value of the two types of consumers, the consumption value of the product.²⁰ Table 3 reports the first-order and second-order polynomial regression estimates for each candidate equilibrium.

Table 3 indicates three important results. First, if a new software product is introduced to the same submarket, the price of the existing software products are lower. Second, software price is increasing in the number of high-type consumers and decreasing in the number of low-type consumers. Lastly, software price is decreasing in the waiting value of high-type consumers, but increasing in the waiting value of low-type consumers.

²⁰The consumption value of a software product is defined by $X_{jt}\psi$, where X_{jt} includes online rating score, product newness and holiday dummies.

Table 3: Polynomial Regression of Pricing Policy Function

State Variables	First-Order Polynomial			Second-Order Polynomial		
	Eqm1	Eqm2	Eqm3	Eqm1	Eqm2	Eqm3
s_1 (# competitors)	-0.358***	-0.406***	-0.403***	-0.403***	-0.078**	-0.001
square				0.009***	0.003***	0.003***
s_2 (# high-type, million)	9.345***	4.431***	6.806***	29.717***	22.150***	26.448***
square				-10.954***	-9.400	-12.389*
s_3 (# low-type, million)	-0.799***	-0.041	-1.634***	1.904***	1.963***	5.050***
square				-0.390***	-0.235***	-0.486***
s_4 (high-type waiting value)	-2.917***	-3.298***	-2.712***	-0.563***	-0.191***	-0.364***
square				-0.068***	-0.132***	-0.176***
s_5 (low-type waiting value)	3.417***	3.940***	3.521***	-2.422***	-2.629***	-2.873***
square				0.246***	0.312***	0.450***
s_6 (its consumption value)	1.448***	1.727***	1.838***	1.294***	1.641***	1.897***
square				-0.012***	-0.025***	-0.041***
R-square	0.56	0.58	0.61	0.65	0.66	0.69

Note: * indicates significance at 10 percent level; ** indicates significance at 5 percent level; and *** indicates significance at 1 percent level.

6 Counterfactuals

In this section, I employ the recovered parameters in the demand and supply model to conduct counterfactual exercises. The goal is to explore such a question often raised in two-sided markets: what are the impacts of platform pricing on the number of consumers, the entry and price of the affiliated products, and their own profitability?

In the video game industry, console platforms have the following price choices. On the one side of the market, console providers charge consumers a one-time payment for access to their platforms, i.e., console price. On the other side of the market, they charge game producers royalty fee which is fixed payment per unit sold to consumers. Besides, game producers also have to spend a one-time payment of research and development if they want to join a platform. Even though this cost does not go to platform providers' pockets, they can still strategically influence this software entry fees either by offering easier-learning development technology or by subsidizing software R&D directly.

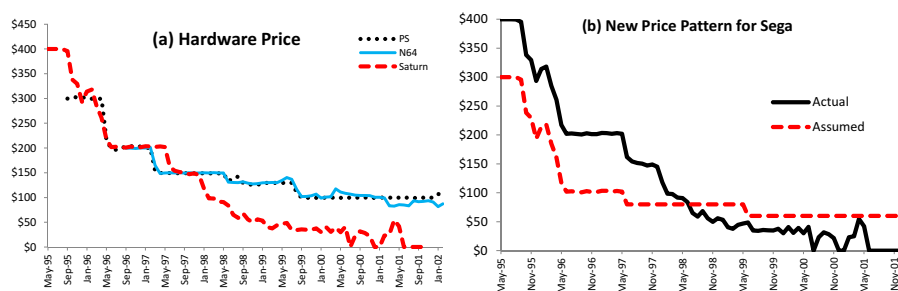
I conduct three counterfactual exercises to focus on the three platform prices, respectively. The first counterfactual examines the benefit of reducing consumer entry fees by assuming that Sega has reduced its console price. The second counterfactual explores the benefit that Sega may have received from subsidizing software R&D. The last counterfactual analyzes the impacts of charges to software producers by assuming that Sega's royalty fees has been \$5 lower per copy. The goal is to

investigate whether Sega Saturn can survive by adjusting any of its price choices.

6.1 Console Price

Figure 6 (a) presents the prices of the three fifth-generation consoles in every month during the sample period. Console prices were generally declining over time. It also shows that Sega's console prices were \$100 higher than its competitors at the same age for the first two year. High console price discourages consumer entry and hence software entry. It is speculated that contributes to the failure of Sega.²¹ Hence, the first counterfactual assumes that Sega has followed the price schedule of its main rival, Sony PlayStation One (see figure 6 b). The goal is to investigate whether Sega could survive if it has charged lower console price.

Figure 6: Hardware and Software Price over Time

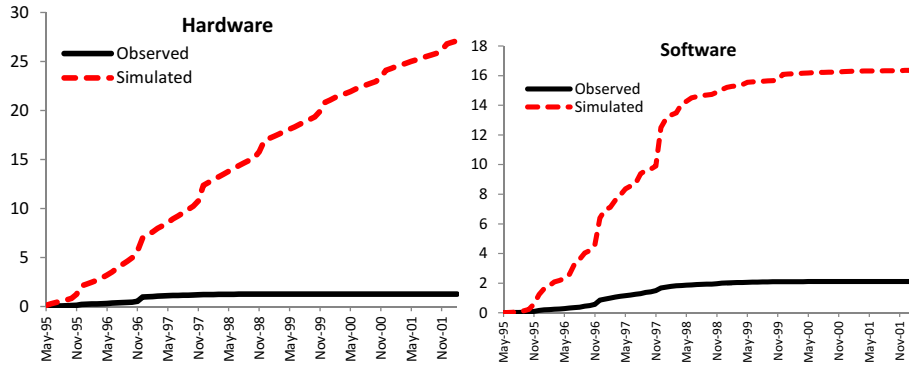


With the proposed price schedule for Sega, I simulate the installed base of users and accumulative sales from sports games for each console. Figure 7 presents the observed vs. simulated values. Clearly, Sega would install a considerable base of users and generate a great amount of sales from sports games. Lower console price attracts more users and hence more games, and then more users and so on. It indicates that Sega Saturn would not be shaken out of the market if it had gotten right the the two-sided business pricing model. On the contrary, its rivals, PlayStation One and Nintendo 64 would suffer from Sega's lower console price strategy. Especially, Nintendo 64's total sales from sports games would shrink to half of the observed number at the end of the sample period.

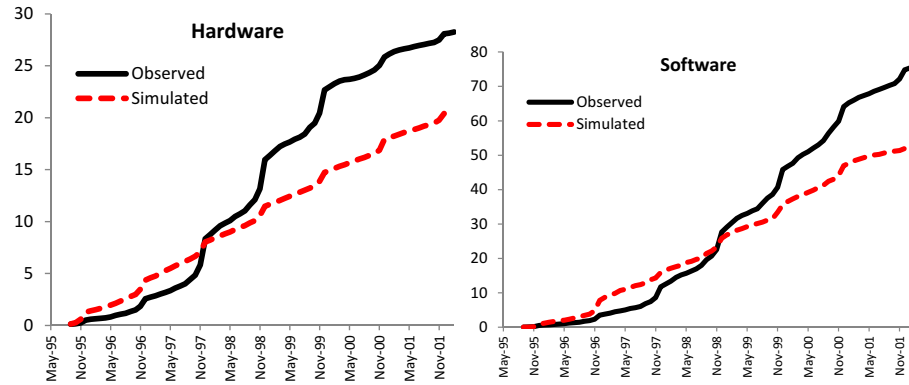
²¹See wikipedia, http://en.wikipedia.org/wiki/Sega_Saturn#cite_note-16.

Figure 7: Installed Base and Accumulative Game Sales (Millions)

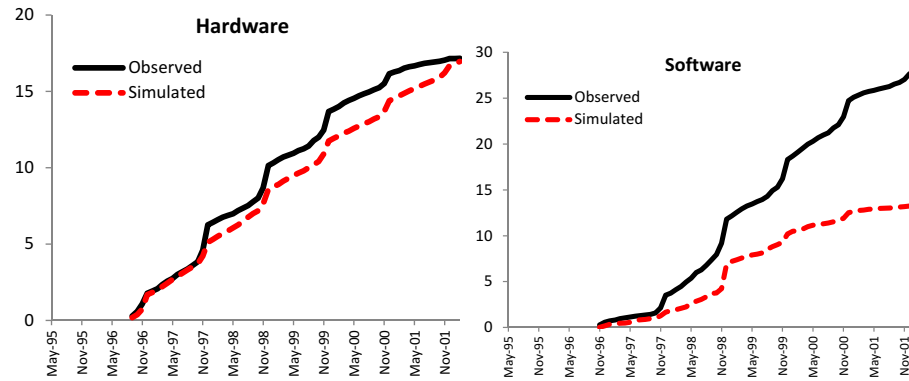
(a) Sega Saturn



(b) Sony PlayStation One



(c) Nintendo 64



6.2 Subsidize Software Entry

Compared to PlayStation, Saturn is difficult to develop for. High R&D cost discourages software entry and thus discourages consumer entry. The second counterfactual analyzes the benefit of subsidizing software entry by assuming that Sega has subsidized new games by \$1M per game.

6.3 Royalty Fees

In a two-sided market with indirect network effects, the effect of price on demand can be larger than in other markets. For instance, the lower software price not only attracts elastic consumers to purchase software, but also lead to higher value attached to the console hardware and thus attract more consumers to come on board. The value extracted from the increased consumers magnifies the value of lowering software price, which leads to a yet bigger price decrease.

In this counterfactual, I assume that Sega's royalty fee is \$5 lower per copy than the estimated value. For software producers, lower royalty fees implies lower unit cost. The equilibrium price depends on the demand elasticity which is more complicated in this industry.

7 Conclusion

This paper develops a framework to study pricing and entry in two-sided markets with positive indirect network effects. It presents a dynamic equilibrium model to describe consumers' dynamic demand for competing hardware platforms and their affiliated software products, and software firms' dynamic pricing and entry.

To estimate the model, it provides a new practical methodology for structural estimation of dynamic equilibrium models.

References

- [1] Aguirregabiria, V., and Mira, P. (2007): “Sequential Estimation of Dynamic Discrete Games”, *Econometrica* 75(1), 1-53.
- [2] Armstrong M. (2006): “Competition in Two-Sided Markets”, *RAND Journal of Economics*, 37(3), 668-691.
- [3] Bajari, P., Benkard, L. and Levin, J. (2007): “Estimating Dynamic Models of Imperfect Competition”, *Econometrica* 75(5), 1331-1370.
- [4] Bajari, P., Hong, H. and Ryan, S. (2010): “Identification and Estimation of a Discrete Game of Complete Information”, *Econometrica* 78(5), 1529-1568.
- [5] Chintagunta, P. and Song, I. (2003): “A Micromodel of New Product Adoption with Heterogeneous and Forward Looking Consumers: Application to the Digital Camera Category”, *Quantitative Marketing and Economics*, 1(4), 371-407.
- [6] Ching, A., Imai, S., Ishihara, M., and Jain, N. (2009): “A Practitioner’s Guide to Bayesian Estimation of Discrete Choice Dynamic Programming Models”, working paper.
- [7] Erdem, T., Imai, S. and Kean, M. (2003): “A model of Consumer Brand and Quantity Choice Dynamics under Price Uncertainty”, *Quantitative Marketing and Economics*, 1(1), 5-64.
- [8] Doraszelski, U. and Satterthwaite, M. (2010): “Computable Markov-Perfect Industry Dynamics”, *RAND Journal of Economics*, 41(2), 215-243.
- [9] Ericson, R. and Pakes, A. (1995): “Markov-Perfect Industry Dynamics: A Framework for Empirical Work”, *Review of Economic Studies* 62, 53-82.
- [10] Gowrisankaran, G. and Rysman M. (2009): “Dynamics of Consumer Demand for New Durable Goods”, manuscript.
- [11] Hagiu, A. (2006): “Pricing and Commitment by Two-Sided Platforms”, *RAND Journal of Economics*, 37(3), 720-737.

- [12] Hendel, I. and Nevo, A. (2007): “Measuring the Implications of Sales and Consumer Inventory Behavior”, *Econometrica* 74(6), 1637-1673.
- [13] Imai, S., Jain, N., and Ching, A. (2009): “Bayesian Estimation of Dynamic Discrete Choice Models”, *Econometrica* 77(6), 1865-1899.
- [14] Jiang, R., Manchanda, P., and Rossi, P. (2009): “Bayesian Analysis of Random Coefficient Logit Models Using Aggregate Data”, *Journal of Econometrics* 149, 136-148.
- [15] Keane, M. and Wolpin, K. (1994): “The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence”, *Review of Economics and Statistics*, 76(4), 648-672.
- [16] Lee, Robin (2010): “Dynamic Demand Estimation in Platform and Two-Sided Markets”, Working Paper
- [17] Nair, H. (2007): “Intertemporal Price Discrimination with Forward-Looking Consumers: Application to the US Market for Console Video-games”, *Quantitative Marketing and Economics*, 5, 239-292.
- [18] Pakes, A. and McGurie, P. (2001): “Stochastic Algorithms, Symmetric Markov Perfect Equilibrium, and the Curse of Dimensionality”, *Econometrica* 69(5), 1261-1281.
- [19] Rochet, Jean-Charles, and Jean Tirole (2006): “Two-Sided Markets: A Progress Report.” *The RAND Journal of Economics*, 35(3), 645-667.
- [20] Rust, J. (1987): “Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher,” *Econometrica*, 55(5), 999–1033.
- [21] Rysman, Marc (2009): “The Economics of Two-Sided Markets.” *Journal of Economic Perspectives*, 23(3), 125-143.
- [22] Weyl, E. Glen (2010): “The Price Theory of Multi-Sided Platforms.” *American Economic Review*, 100(4), 1642-1672.

Appendix A: Figures

Appendix B: Competition Structure of Software Market

In section 3, I assume that products compete within a submarket and submarkets are separate from each other. Below I specify three different regression models to test the substitution between sports games. Table 4 presents the empirical results which are consistent with that assumption.

Table 4: Empirical Results of Testing Software Competition Structure

	Model 1 price (\$)	Model 2 $\ln(q_{jt})$	Model 3 $\ln(q_{jt})$
its own price		-.0126*** (.0019)	-.009*** (.002)
competition in the same submarket	-.148** (.060)	-.0359*** (.0045)	-.219*** (.019)
competition from other submarkets	-.011 (0.066)	.0002 (.0004)	-.012 (.024)
online rating score	1.417*** (.047)	.3419*** (.0099)	.261*** (.010)
product age (months)	-1.141*** (.015)	-.1988*** (.0099)	-.199*** (.004)
age square	.013*** (.002)	.0015*** (.0000)	.002*** (.000)
market size (million)	3.353*** (.088)	.2438*** (.0466)	.803*** (.034)
R-square	0.68	0.63	0.53
observations	13779	13024	12794

Note: * indicates significance at 10 percent level; ** indicates significance at 5 percent level; and *** indicates significance at 1 percent level.

In the first regression model, the dependent variable is the product price and the independent variables include: (i) the competition level within a market measured by the number of competitors in the same market; (ii) the competition from other markets measured by the number of products in all other markets; (iii) observed product characteristics including the online rating score and the product age measured by the months after release; (iv) the market size measured by the log of the installed base of users of the compatible console; and (v) monthly dummy. The second column in table 4 lists the results of this regression. The first important result is that the price is lower by \$0.148 if additional game is released in the same market and this impact is significant at 5% significance level. It implies that the competition within a market is strong. The second important

result is that the competition effect from other markets is not significant, which implies that markets are separate from each other. Besides, price of a product is increasing in its online rating score, declining in its age and increasing in the installed base of the compatible console.

In the second regression model, the dependent variable is the log of the unit sales (measured in thousands). The independent variables are the same as in the first model except that I include an extra independent variable, the current price. To address the endogeneity of the price, I use the lagged price as an instrument for the current price. The third column in table 4 lists the estimation results which are consistent with the assumption of strong competition within a market and weak competition across markets.

The last regression model mimics Nair (2007).²² I still use the log of the unit sales as the dependent variable. However, I use the log of the unit sales of other products in the same market to measure the competition level within a market, and the log of the unit sales of all products in other markets to measure the competition effect from other markets. To address the endogeneity problem, I use the lagged price as an instrument for the current price, the number of competitors within the a market as an instrument for the within-market sales, and the number of products in other markets as an instrument for the outside-market sales. The results are listed in the last column in table 4. It also shows that the substitution effect within a market is strong while the substitution from the products sold in other markets is insignificant.

Appendix C: Computation of the Jacobian Matrix

The Jacobian Matrix in equation (13) is

$$J_{(\mathbf{q}_{mt} \rightarrow \xi_{mt})} \equiv \|\nabla_{\xi_{mt}} \mathbf{q}_{mt}\| = \begin{bmatrix} \partial q_{1t}/\partial \xi_{1t} & \dots & \partial q_{1t}/\partial \xi_{Jt} \\ \vdots & \ddots & \vdots \\ \partial q_{Jt}/\partial \xi_{1t} & \dots & \partial q_{Jt}/\partial \xi_{Jt} \end{bmatrix}$$

where

$$\frac{\partial q_{jt}}{\partial \xi_{lt}} = \begin{cases} -\sum_{i=1}^I n_{mit} \left[b_{ijt} b_{ilt} + \beta_c b_{ijt} b_{i0t} \frac{\partial EG_{mit+1}}{\partial \xi_{lt}} \right] & \text{if } l \neq j, \\ \sum_{i=1}^I n_{mit} \left[b_{ijt} (1 - b_{ijt}) + \beta_c b_{ijt} b_{i0t} \frac{\partial EG_{mit+1}}{\partial \xi_{jt}} \right] & \text{if } l = j. \end{cases}$$

²²In Nair (2007), the dependent variable is $\ln(s_{jt}/s_{0t})$, where s_{jt} is the market share of game j and s_{0t} is the share of the outside good. He uses $\ln(s_{jt|g}/s_{0t})$ to measure the effect within a market, where $s_{jt|g}$ is the share of units sales of the game within its genre, g . He finds that the substitution effect from other games with the same game genre is not significant, and thus he concludes that video games are separate from each other.

I compute the $\partial G_{mit}/\partial \xi_{jt}$ for $\forall j \in J_{mt}$ as follows. Let $b_{im_0t} = 1 - \sum_{j \in J_{mt}} b_{ijt}$ denote the probability of not purchasing. Notice that $\partial G_{mit}/\partial \xi_{jt}$ and $\partial b_{im_0t}/\partial \xi_{jt}$ are determined by the following system of equations:

$$\begin{cases} \frac{\partial EG_{mit+1}}{\partial \xi_{jt}} = \sum_l \frac{\partial EG_{mit+1}}{\partial n_{mlt+1}} n_{mlt} \frac{\partial b_{im_0t}}{\partial \xi_{jt}} & \text{for all } i \\ \frac{\partial b_{im_0t}}{\partial \xi_{jt}} = -b_{ijt} b_{im_0t} + \beta_c b_{im_0t} (1 - b_{im_0t}) \frac{\partial EG_{mit+1}}{\partial \xi_{jt}} & \text{for all } i \end{cases}$$

In the application part, I only assume two types of consumers. So, the above system includes four linear equations and four unknowns. It is not hard to solve for $\partial G_{mit}/\partial \xi_{jt}$ for all i .

Appendix D: Best Response in Price

The incumbent firm's optimization problem is to pick a price to maximize the discounted profit:

$$\max_{p_{jt}} \pi_j(\tilde{p}_{jt}, p_{-jt}, \mathbf{s}_{mt}) + \beta_f E [V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, \tilde{p}_{jt}, \Psi_{-j}],$$

where

$$\begin{aligned} \pi_j(p_{jt}, p_{-jt}, \mathbf{s}_{mt}) &= (p_{jt} - c) \left[\sum_i n_{mit} b_{ij}(p_{jt}, p_{-jt}, \mathbf{s}_{mt}) \right] \\ b_{ij}(p_{jt}, p_{-jt}, \mathbf{s}_{mt}) &= \frac{\exp(\mathbf{X}_{jt}\psi - \varphi_i p_{jt} + \xi_{jt})}{\exp[\beta_c E B_i(\mathbf{p}_{mt+1}, \mathbf{s}_{mt+1} | p_{jt}, p_{-jt}, \mathbf{s}_{mt})] + \sum_{l \in J_{mt}} \exp(\mathbf{X}_{lt}\psi - \varphi_l p_{lt} + \xi_{lt})}. \end{aligned}$$

Unfortunately, we do not know the analytical function form of the expected values as a function of current price. First, consider the expectation of incumbent value function, $E [V_j(\mathbf{s}_{mt+1} | \Psi) | \mathbf{s}_{mt}, \tilde{p}_{jt}, \Psi_{-j}]$, where the state vector \mathbf{s}_{mt} includes the number of competitors in the same market, the number of active high-type consumers, the number of active low-type consumers, the value of the No. 1 product in the same market, its own value, and the number of time periods to the end. Notice that given the competitors' prices and entrants' entry probabilities, current price only affects the number of next-period active consumers but not other next-period state variables. The number of next-period active consumers is the sum of the number of consumers who do not make any purchase today and the number of new consumers, that is,

$$n_{mit+1} = n_{mit} b_{m_0it} + q_{it}^h,$$

where b_{m_0it} is the probability of not buying any product in market m at time t , $b_{m_0it} = 1 - \sum_{j \in J_{mt}} b_{ij}(p_{jt}, p_{-jt}, \mathbf{s}_{mt})$. Given other agents playing the equilibrium strategies, we can know the analytical function form of $\mathbf{s}_{mt+1}(p_{jt}, p_{-jt}, \mathbf{s}_{mt})$. Moreover, along the estimation procedure, we approximate the value functions $V_j(\mathbf{s}_{mt+1})$ by using second-order-polynomial in state variables. Therefore, we can pin down how current price affects the expectation of incumbent value function. A similar approach can be applied to the expectation of consumer value function.