Failure to Launch in Two-Sided Markets: A Study of the U.S. Video Game Market

Zhou, Yiyi

The State University of New York at Stony Brook

16 October 2012

Failure to Launch in Two-Sided Markets: A Study of the U.S. Video Game Market*

Yiyi Zhou

October 2012

Abstract

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. Over time, this process can lead to a death spiral. This paper develops a dynamic structural model of the video game market to study launch failures in two-sided markets. The paper models consumers' purchase decisions for hardware platforms and affiliated software products and software firms' entry and pricing decisions. This paper also develops a Bayesian Markov Chain Monte Carlo approach to estimate dynamic structural models. The results of the counterfactual simulations show that a failed platform could have survived if it had lowered its hardware prices and that it could not have walked out of the death spiral if it had subsidized software entry.

Keywords: Bayesian Markov Chain Monte Carlo (MCMC) Estimation, Failure to Launch, Two-Sided Market, Indirect Network Effect, Forward-Looking Consumer, Video Game Market

*Department of Economics, The State University of New York at Stony Brook. yiyi.zhou@stonybrook.edu.
This paper is based on my 2012 dissertation. The previous version of the paper was circulated under the title "Bayesian Estimation of a Dynamic Equilibrium Model of Pricing and Entry in Two-Sided Markets: Application to Video Games". I am extremely indebted to my committee, Steven Stern, Simon Anderson, and Federico Ciliberto for their generous guidance, encouragement, and support. The paper has benefited from comments of seminar participants at Florida State, Maryland, Ohio State, Pennsylvania, Stony Brook, Virginia and IIOC meetings, and conversations with Andrew Ching, Hulya Eraslan, Nathan Larson, Andriy Norets, Regis Renault, John Rust, Holger Seig, Edward Vytlacil, Glen Weyl, Kenneth Wolpin, and Daniel Xu. Furthermore, I thank Wei-Min Hu for assistance in acquiring data. I am grateful for financial support from the University of Virginia's Bankard Fund for Political Economy. All errors are my own.

1 Introduction

In many two-sided or "platform" markets, consumers join a platform to access goods that firms affiliated with that platform provide, 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. The success of the affiliated products depends on the number of consumers on the platform. In literature on two-sided markets, this interdependence, or externality between two groups of agents that a platform serves, is called indirect network effects. Moreover, platform markets are often inherently dynamic environments due to the durability of platform intermediaries and the affiliated products. Two-sidedness and dynamics are important features of many key industries such as eReaders and ebooks, video games and consoles, operation systems and software, DVD players and DVDs, and smart phones and apps.

Some platforms may be able to grow rapidly from a small base because customers on one side attract customers from the other side, but most platforms do not. Many banks launched credit card systems in the 1950s, and almost all failed. Sony Betamax lost in the videotape format war with its competitor VHS in the late 1970s and the 1980s, but Sony Blu-ray took the lead over its main competitor, HD-DVD, only one and half years after its launch. This paper asks why some platforms launch successfully but others fail.

Theory tells us that platforms need to "get both sides on board" to launch successfully (Rochet and Tirole, 2003; Armstrong, 2006; Hagiu, 2006; Weyl, 2010). In two-sided markets, pricing on one side of the market not only affects the demand on that side but also affects participation on the other side of the market. Hence, charging low, or even negative, prices during the launching stage is crucial to achieve the snowball effect. In practice, Amazon sold the Kindle Fire slightly below its manufacturing cost to attract users during the launching stage[1], and yellow-page publishers offer free advertisements in the first year that they enter

[1] According to an IHS analysis, Amazon's Kindle Fire (8GByte) costs $201.70 to manufacture but was sold at $199 at release.

a local market.

To analyze how a platform grows or shrinks, we need to know how customers on both sides of the market behave. In this paper, I set up a dynamic structural model that describes consumers' purchase decisions on hardware platforms and their affiliated software products, and software firms' dynamic pricing and entry decisions.[2] To estimate the model I use a data set from the 32- and 64-bit generation, or fifth-generation, U.S. video game market including three dominating consoles: Sega Saturn, Sony PlayStation, and Nintendo 64. Sega Saturn failed during this generation, even though it had been very successful in the previous generation. The counterfactual simulations suggest that Sega priced inconsistently with the two-sided business pricing model and therefore was shaken out of the market. Sega would have survived if it had lowered its console price to attract more consumers and hence more games. However, it would not have walked out of the death spiral if it had only subsidized software entry.

This paper contributes to literature on two-sided markets that has been growing quickly in the last decade. Rysman (2009) provides a general review of the literature in this field. In those markets with positive indirect network effects, one side of the market is always waiting for the other side to act before taking its own action. Previous literature has emphasized that platforms need to "get both sides on board" and solve the "chicken-and-egg" coordination problem that Caillaud and Jullien (2003) originally pointed out. With a few exceptions, previous studies have usually modeled the launch of new platforms as an event, not a process; they have not focused on the start-up problems that new platforms face.[3] This paper analyzes

[2]I do not model platform makers' decisions on price and entry for two reasons. First, both consumers and software firms are modeled as forward-looking agents, and thus their decision-making processes are complicated by themselves. It is extremely hard to go further to model the decisions of platform makers who choose their price and entry taking into account consumers' purchase decisions and software firms' price and entry decisions. Second, the goal of this paper is to study launch failures in two-sided markets, in particular whether a failed platform would survive by taking alternative strategic options. To achieve this goal, I model how the two sides respond to platform makers' choices and simulate the results when a failed platform takes an alternative option.

[3]One exception is Evans and Schmalensee (2010), who show that a platform business needs to pass an initial critical mass that depends on the nature of network effects, the dynamics of customer behavior, and the distribution of customer tastes.

the dynamics of platform growth and looks at how a price change during the launching stage affects a platform's formation process.

In this paper, consumers are heterogeneous, forward-looking, and have rational expectations about future software entry and prices. In each time period, consumers choose whether and when to purchase hardware and affiliated software. The hardware purchase and the software purchase are interdependent decisions. On one hand, the value of hardware depends on the value of being able to purchase affiliated software, so consumers rationally anticipate the software market when they make their hardware purchasing decisions. On the other hand, the number of potential consumers for a software product depends on how many consumers have purchased the compatible hardware. On the software side of the market, there exists a finite number of separate submarkets. In each submarket and each time period, the existing software firms decide how much to charge, and potential entrants decide whether to enter. At equilibrium, given other agents' strategies, each agent's best response is the solution to a single-agent dynamic programming problem. Furthermore, the equilibrium is the fixed point of the system of best response operators.

To estimate this complicated model, this paper provides a practical estimation procedure that combines the Bayesian algorithm and the fixed-point algorithm. In the outer-loop, I use the Metropolis-Hastings algorithm to draw a sequence of parameter vectors from their posterior distributions. In the inner-loop, for a given parameter vector along the MCMC chain, I non-parametrically approximate each agent's value function and best response function by using the pseudo-value functions and pseudo-best response functions from previous MCMC iterations. Then I adopt an interpolation approach to obtain each agent's continuation value, solve for each agent's best response function (pseudo-best-response function) and value function (pseudo-value function) given that other agents play their equilibrium strategies, and store these pseudo-best response functions and pseudo-value functions for future MCMC iterations. This estimation procedure does not fully solve the dynamic model but incorporates the approximation and the interpolation approaches. The estimation procedure significantly alleviates computational burden and makes the Bayesian MCMC method applicable to estimating dynamic equilibrium models.

This paper contributes to the literature on Bayesian estimation methods that has been commonly applied to static discrete choice models with latent variables.[4] Imai, Jain and Ching (2009), and Norets (2009) pioneered the use of Bayesian estimation methods for dynamic discrete choice models. In contrast to those two papers, the estimation procedure in this paper is designed to estimate dynamic games that are more complicated because the equilibrium is the fixed point of the best response system. This paper also extends the estimation method of Pakes and McGuire (2001) to the Bayesian framework. In Pakes and McGuire's algorithm, they approximate the continuation value using the average of the returns from past outcomes of the algorithm, and the value and policy functions are updated at a recurrent class of points, rather than at all possible points, in the state space.

The rest of the paper is organized as follows. Section 2 describes the data set and the U.S. video game industry. Section 3 builds a structural model of dynamic demand and dynamic supply. Section 4 describes the Bayesian MCMC estimation procedure and discusses the related computational issues. Section 5 reports the estimation results and examines the fit of the model. In Section 6, I conduct two counterfactual exercises to examine Sega's alternative strategic options. Section 7 concludes the findings.

2 The U.S. Video Game Market

Since Pong was first introduced in the early 1970s, the U.S. video game industry has grown significantly. In 2008 the industry grossed \$22 billion, more than twice the total box-office revenue in the movie industry, which grosses \$10 billion. The video game industry is a two-sided market in which consoles (hardware) act as platform intermediaries, and consumers and producers of video games (software) are on the two sides of the market. On one side of the market, console providers design and sell consoles to consumers who pay a one-time

[4] See Albert and Chib (1993), McCulloch and Rossi (1994), Jiang, Manchanda and Rossi (2009).

fixed fee for the console that allows them to join a platform. On the other side of the market, console providers charge independent game producers a royalty fee for the rights to the code that allows game producers to make their games compatible with the console. The royalty fee is not a one-time payment; rather, it is a unit payment for each copy that they sell to consumers. In fact, console providers manufacture all the video games themselves so they can track sales for royalty collection. Console makers also develop and publish video games for their own hardware platforms; in-house game titles do not need to pay royalty fees to console makers. I treat their prices and availabilities as given to other independent software firms.

To satisfy consumers' needs for the latest technology, console providers have introduced new systems approximately every five years. The data used in this paper cover the 32- and 64-bit generation, or fifth generation, of the U.S. video game market. The data include three specific consoles: the Sega Saturn, released in May 1995; the Sony PlayStation, released in September 1995; and the Nintendo 64, released in September 1996. One novelty of this generation is that Sega, a very successful incumbent for many years in this industry, failed to launch its platform and exited the industry. Additionally, none of the consoles was backwardly compatible, eliminating the concern that a previously existing consumer base might have given one console platform an advantage.

2.1 Data

The main data set is obtained from the NPD Group, a market research firm. The data include the monthly revenue and unit sales of three fifth-generation consoles, Sega Saturn (Saturn), Sony PlayStation (PS), and Nintendo 64 (N64), from May 1995 through February 2002. Sony was a new entrant to this industry and the PS soon became the leading platform, taking around 60 percent of the market. Nintendo was the main competitor of Sony and had a market share of 37 percent. Sega ran a distant third behind the other two and actually stopped producing in 1998. I take the ratio of revenue over unit sales in each month to calculate the console price. Since the sixth generation started when Sony launched its PlayStation 2 in October 2000, the data set covers the entire fifth-generation video game industry.

The data set also includes the monthly revenue and unit sales for 1,697 unique game titles released for the three consoles during this period: 240 Saturn game titles, 1,172 PS game titles, and 385 N64 game titles. The data set was collected from 30 of the largest retailers in the U.S., retailers that account for around 85 percent of video game sales. The NPD Group extrapolated the set for the entire U.S. market. I take the ratio of revenue over unit sales in every month to calculate the game price. The data I use to estimate the game market only includes sports games. I did this because it is relatively easy to sort sports games into groups, and using a smaller sample reduces estimation time. The data used in the estimation contains 397 sports games divided in 29 software submarkets. Additionally, I collected the data on user and critics rating scores for each game title from several large 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 the U.S. market for the Saturn, PS, and N64 were 1.28 million, 28.25 million, and 17.17 million, respectively. The total unit sales of their affiliated video games were 8.09 million, 300.02 million and 111.55 million, respectively. Even though Saturn was the first mover, the console became the "other" system barely two years after its release, running a distant third behind its two rivals.

2.2 Industry Description

Below I briefly discuss the important features of this industry, the positive indirect network effect, the declining pattern of game price and sales, and the seasonality of console and game sales.

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 (million)	1.28	28.25	17.17
SOFTWARE			
Total active titles	240	1172	385
Total unit sold (million)	8.09	300.20	111.55
Average units sold per title (million)	0.03 (0.04)	0.26 (0.48)	0.39 (0.67)
Average revenue per title (million)	1.25 (1.61)	8.47 (26.71)	18.73 (34.69)
Average starting price	\$52.66 (\$7.83)	\$41.57 (\$12.02)	\$54.57 (\$8.16)

Notes: Summary statistics for Saturn are for the 82-month period between May 1995 and February 2002; statistics for PS are for the 78-month period between September 1995 and February 2002; and statistics for N64 are for the 66-month period between September 1996 and February 2002. Ending price, Installed base, total active titles and total unit sold with any console are for February 2002, the last month in the sample. Numbers in parenthesis are standard deviations. Data source: NPD group.

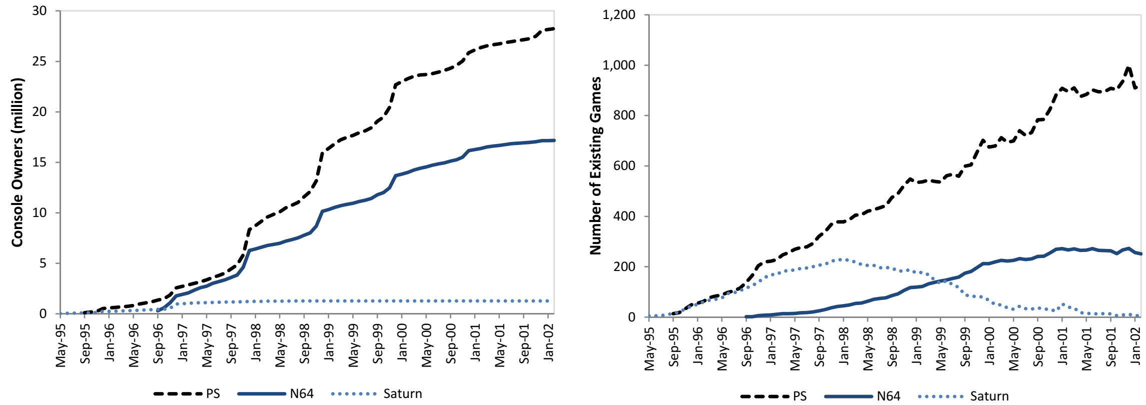
1. Positive Indirect Network Effects

Consumers buy a console to access its video games, and game producers make their games compatible with a console to reach consumers who own that console. The number of users of a console is therefore largely contingent on current and expected availability and game prices, and the number of games affiliated with a console depends on how many users have purchased and are expected to purchase that console. Figure 1 shows that the installed base of hardware and the software variety have the same growth pattern, implying positive correlation between consumer entry and software entry.

On one side of the market, consumers decide whether to purchase consoles and games. A console has no stand-alone value; its value comes from its compatible game titles. Figure 1 (a) presents the number of each console's owners during the sample period. The installed

Figure 1: Hardware Installed Base and Software Variety

(a) Hardware Installed Base (b) Active Game Variety



Notes: (a) The installed base is measured by the accumulative units sold of each console in millions. The monthly sales of Saturn were below 0.5 million units after January 1997. (b) Active games are referred to those which has positive sales.

bases of PS users and N64 users grew quickly during this period. However, the number of Saturn owners stopped growing one and a half years after its release.

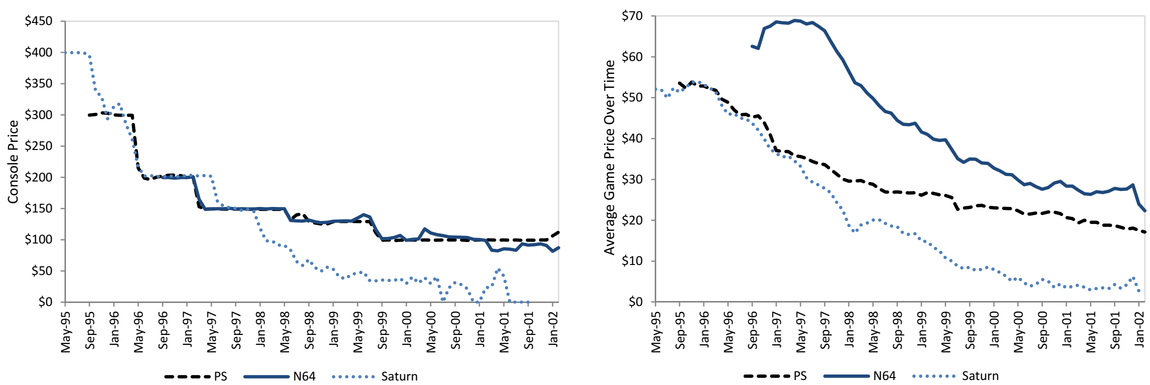
On the other side of the market, incumbent game producers choose their prices, and potential entrants choose whether to enter the market. Figure 1 (b) presents the number of existing game titles sold for each console in every month during the sample period. The number of PS game titles and the number of N64 game titles grew quickly. In contrast, the number of Saturn game titles started to shrink from January 1998.

2. Console Price and Game Price Decline over Time

Console prices are shown in Figure 2(a). Saturn started retailing for $399 but in September 1995 cut its price by $100 to match the price of the newly launched PS. PlayStation started at $299 in September 1995 and suddenly dropped below $200 in May 1996 before N64 was launched. Nintendo 64 was sold at $199 when it came to market and thereafter was sold at almost the same price as PS. Both PS and N64 cut their prices by $50 in March 1997, by around $20 in June 1998, and by around $30 in September 1999. Overall, hardware prices were declining. It is widely speculated that all the major consoles were initially sold at a price near marginal cost. Industry reports also indicate that console prices fell slower than production costs, and thus the margin actually increased over time.

Figure 2(b) shows the average game price for each console over time. Software prices increased slightly during the first few months after a console was introduced, and thereafter declined smoothly over time. Initially, N64 games were much more expensive than others, but the price gap became smaller over time.

Figure 2: Console Price and Average Game Price over Time



(a) Hardware Price (b) Average Price of Games

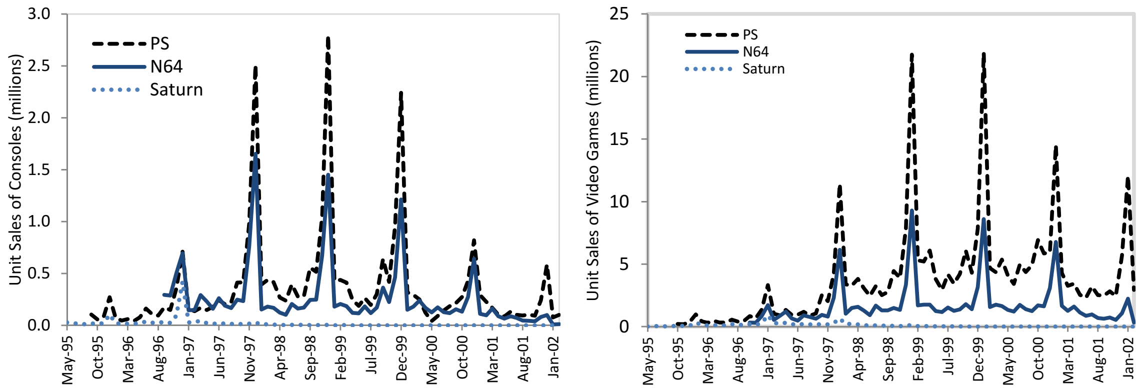
Notes: (a) Average monthly (nominal) prices faced by consumers in retailer stores for each console in the U.S. market from May 1995 to February 2002. (b) Average monthly (nominal) prices of video games released for each console.

3. Seasonality and Life-Cycle Pattern

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

After adjusting for seasonality, both the console and game sales had U-shaped patterns; that is, both grew initially until reaching a peak and thereafter declined. This life-cycle pattern can be explained as follows. In the early months, very few consumers owned consoles and the small market size resulted in low game sales. Meanwhile, very few games were available, so the consumption values of consoles were low, and console sales were low. However, as more consumers purchased consoles over time, game sales increased. Meanwhile, as more and more new games were released over time, the consumption value of consoles increased and console sales increased. At the end of the sample period, the new-generation consoles and games were available, so the sales of old-generation consoles and games declined over time.

Figure 3: Unit Sales of Consoles and Games (millions)



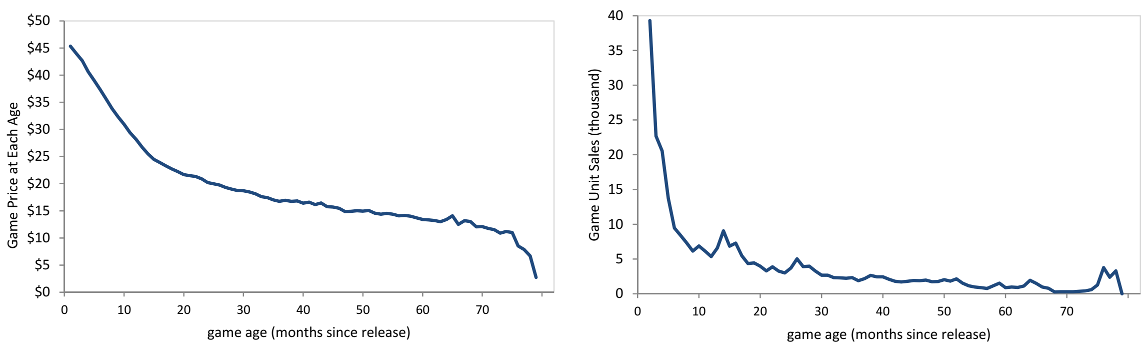
4. Game Prices and Sales Decline with Age

An important feature of the video game market is that game price and sales start at a high level then decline rapidly in the first six months after being released. In Figure 4, the horizontal axis is the game age measured by the months since introduction, and the vertical axis is the average game price in (a) and the average unit sales in (b). The average game price was around \$45 per copy at release and then dropped to about \$23 the following year. The average game unit sales were around 40,000 in the first month and then fell to around 5,000 per month after the first year.

What drives game prices and sales to drop so quickly? A falling-cost explanation is not convincing for this industry. Once a video game is developed, the producer only needs to pay royalty fees to the console maker and pay for its own production cost. Both costs

Figure 4: Game Price and Unit Sale at Each Age

(a) Game Price (b) Game Unit Sales



remain roughly constant per unit over time.[5] The most reasonable explanation is intertemporal price discrimination (Nair 2007). Consumers are heterogeneous in their preferences for product characteristics, price, or both. Consumers purchase consoles and games at different times, and, as a result, the distribution of potential buyers of a game title changes over time. The different composition of consumers at different times induces game producers to charge different prices. Intuitively, consumers with high net valuations purchase earlier than those with low net valuations. Thus, it is optimal for game producers to set high initial prices to sell to consumers with high net valuations and then cut prices to appeal to consumers with low net valuations. Additionally, the entry of new games leads to more-intense competition and thus induces the manufacturers of existing game titles to cut their prices.

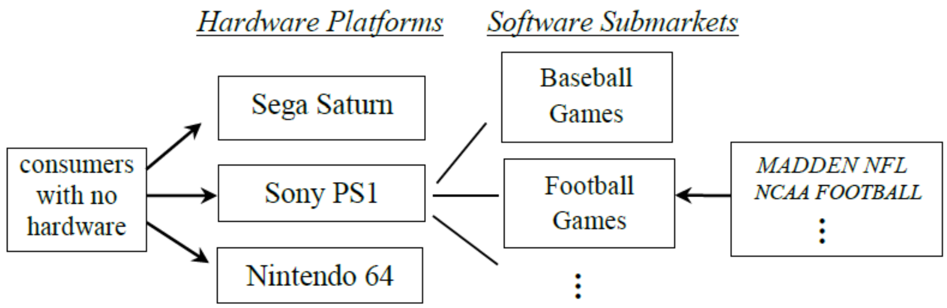
3 Model Framework

In this section, I present a structural model to describe consumers' demand for hardware and affiliated software and software firms' choices of entry and prices. The model is dynamic, time

[5]Coughlan (2001) reports that production and packaging costs for 32-bit CD-ROM games remains roughly constant at $1.50 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.

is discrete. and the horizon is finite.[6] There exists a finite number of hardware platforms. Platforms' choices, including the entry fees to each side and the transaction fee, are taken as given at the beginning of the first time period.[7] The structure of the model can be displayed by the Figure 5.

Figure 5: Model Structure



On the consumer side, consumers with no hardware decide whether to buy one in each time period. Each consumer is allowed to buy at most one hardware in her life-time.[8] Once she owns one hardware platform, she become a potential buyer for the affiliated software products. The software side consists of a finite number of separate submarkets. Each consumer can purchase at most one software product within a submarket. This setup of the software market explicitly assumes that software products in the same submarket are substitutable and that software products from different submarkets are independent. In the context of video games, 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.[9] I examine the

[6]In the application to the video game industry, I focus on the 5th generation. I assume that this generation dies after 100 months (roughly 8 years).

[7]The model does not endogenize the platforms' choices. Rather, it describes how the consumers and the software firms respond to platforms' choices. This can be treated as a two-stage game: in the first stage, platforms choose their prices to consumers (console prices) and the entry cost to software firms before the generation starts; and, in the the second stage, with all the choices made by the platforms given, consumers make their purchase decisions of hardware and affiliated software, and software firms decide on whether to enter and what prices to charge. The model can be considered as the second stage of the two-stage game.

[8]Ruling out multiple console purchasing may potentially cause biases. This paper does not allow for consumer multi-homing for two main reasons. First, including multi-homing purchase significantly complicates the estimation. Lee (2010) allows for multi-homing, 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 multi-homing is unavailable.

[9]For example, PS Football games is a submarket, PS Baseball games is a submarket, Nintendo Football

substitutability of software products (see Appendix A for details). The preliminary empirical results indicate that games grouped in the same submarket are strongly substitutable, while games grouped in different submarkets are weakly substitutable.

On the software side, each software firm is assumed to produce only one product.[10] The following events occur in each software submarket and in each time period:

(i) Each incumbent software firm decides how much to charge. Each potential entrant draws an entry cost from a known distribution, and decides whether to enter. If it enters, it starts to earn profit in the next period. Price and entry decisions are made simultaneously.

(ii) Potential buyers immediately observe the software prices but not the entry outcomes. However, they have rational expectations about software firms' entry strategy. They decide whether to buy an affiliated software product and, if so, which one. Once she makes a purchase in a submarket, she leaves that submarket forever.

(iii) Software entry decisions are implemented. We move to the next period.

Below, I first describe consumer dynamic purchase of hardware and software, software firms' dynamic pricing and entry, and lastly the equilibrium concept for the model.

3.1 Demand for Hardware

There is 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. A hardware product itself has no stand-alone value; its value comes from the affiliated software. Let Γ_{ilt} be the expected value of optimally purchasing software associated with platform l. The functional form of Γ_{ilt} is derived from the software adoption portion of the model, which will be described in the next subsection. The expected lifetime utility that a

games is another submarket, and so on.

[10] In reality, some software firms publish more than one software titles. For example, EA Sports published more than 100 game titles from May 1995 to February 2002. However, it is computationally difficult to accommodate multi-product firms. This single-product assumption holds if the team of publishing a software title is an independent decision maker and thus each team can be treated as a single software firm.

type-i consumer can obtain from purchasing platform l at time t is

$$U_{ilt} = \Gamma_{ilt} - \alpha_i P_{lt} + X_t\gamma + \zeta_{lt} + \varepsilon_{ilt},$$

where P_{lt} is the price of hardware product l, α_i represents consumer type-specific sensitivity to price, X_t is the holiday dummy[11], ζ_{lt} represents additional hardware characteristics observed by consumers but not by researchers and ε_{ilt} is idiosyncratic consumer taste.

Since hardware products are durable goods, consumers are forward-looking when they decide whether to buy them. The no-purchase option captures the 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:

$$U_{i0t} = \beta_c E_t \left[max\{\underset{l}{max} U_{ilt+1}, U_{i0t+1}\} \right] + \varepsilon_{i0t},$$

where β_c is the consumer's discount factor and the expectation is taken with respect to the distribution of future variables unknown to the consumer conditional on the current information. As usual in the literature, ε_{ilt} and ε_{i0t} 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 affects consumer purchase decision of hardware at time t. Then, a type-i consumer's dynamic optimization problem can be written as

$$H_{it}(\varepsilon_{it}, \mathbf{S}_t) = max \left\{ \underset{l}{max}\ U_{ilt},\ \varepsilon_{i0t} + \beta_c E\left[E_\varepsilon H_{it+1}(\varepsilon_{it+1}, \mathbf{S}_{t+1}) \mid \varepsilon_{it}, \mathbf{S}_t\right] \right\},$$

where $H_{it}(\varepsilon_{it}, \mathbf{S}_t)$ is type-i consumer's value function with information set $\mathbf{S}_t$ and tastes ε_{it}. Let $H_{it}(\mathbf{S}_t)$ denote the expected value function, that is, the value function before consumers

[11] It includes two variables, $Nov_t = I\{Novermber\}$ and $Dec_t = I\{December\}$, where $I\{\cdot\}$ is an indicator function. Hence, $X_t\gamma \equiv \gamma_{Nov} Nov_t + \gamma_{Dec} Dec_t$.

know their demand shocks ε_{it},

$$H_{it}(\mathbf{S}_t) = \int_{\varepsilon} H_{it}(\varepsilon_{it}, \mathbf{S}_t)) dF_{\varepsilon}(\varepsilon_{it}).$$

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

$$H_{it}(\mathbf{S}_t) = ln\{\sum_l exp(\Gamma_{ilt} - \alpha_i P_{lt} + X_t\gamma + \zeta_{lt}) + exp[\beta_c EH_{it+1}(\mathbf{S}_{t+1} \mid \mathbf{S}_t)]\}. \quad (1)$$

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

$$B_{ilt}(\mathbf{S}_t) = \frac{exp(\Gamma_{ilt} - \alpha_i P_{lt} + X_t\gamma + \zeta_{lt})}{exp[\beta_c EH_{it+1}(\mathbf{S}_{t+1} \mid \mathbf{S}_t)] + \sum_l exp(\Gamma_{ilt} - \alpha_i P_{lt} + X_t\gamma + \zeta_{lt})}. \quad (2)$$

The demand for the hardware l at time t is

$$Q_{lt} = \sum_i N_{it} B_{ilt},$$

where N_{it} is the number of consumers who have not purchased any hardware product at time t. Recall that a consumer is assumed to buy at most one hardware in her life time, and once she makes a purchase of hardware, she is no longer an active consumer for the hardware market. Hence, in this dynamic models of discrete choice demand, $\{N_{it}\}_{t=1}^{T}$ evolves according to

$$N_{it+1} = N_{it}(1 - \sum_l B_{ilt}).$$

3.2 Demand for Software

Recall that the software market consists of a finite number of separate submarkets. Below I describe consumers' demand for software, and software firms' pricing and entry in a representative software submarket.

Software Utility

Let J_{mt} denote the set of software products available for consumers to purchase in submarket m at time t. A type-i consumer's lifetime expected utility from purchasing a software product $j \in J_{mt}$ at time t (provided she already owns the compatible hardware) is

$$u_{ijt} = x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt} + \epsilon_{ijt},$$

where x_{jt} is a vector of observed software product characteristics, including platform-specific dummy, online rating score, product age, and holiday dummies;[12] p_{jt} is the price of software j; ξ_{jt} is additional software characteristics observed by consumers but not by researchers; and ϵ_{ijt} is idiosyncratic consumer taste. Here, ψ represents consumer preferences in observed software characteristics, and φ_i is type-i consumer's sensitivity to software price.

In the dynamic environment, the utility of not buying in the submarket m at time t is the sum of the discounted expected value of waiting and an idiosyncratic consumer taste:

$$u_{im_0t} = \beta_c E_t\left[max\{\underset{j \in J_{mt+1}}{max} u_{ijt+1}, u_{im_0t+1}\}\right] + \epsilon_{im_0t}$$

where ϵ_{im_0t} is the idiosyncratic taste from not buying any product in submarket m. 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 previous research on estimating dynamic demand models assumes that consumer purchase decisions are only based on a scalar state variable (the inclusive value) which follows

[12] Consumers' utility declines with game age in different ways for new games and old games. So, I treat a game as a new game if it has been in the market shorter than one year, and as an old game if it has been in the market longer than one year. Hence, $x_{jt}\psi = \psi_{N64}I\{j\, is\, a\, N64\, game\} + \psi_1 rating_j + \psi_2 min(age_{jt}, 12) + \psi_3 max(age_{jt} - 12, 0) + \psi_{Nov}Nov_t + \psi_{Dec}Dec_t$, where age_{jt} is the months after release.

an AR(1) process.[13] Such a restriction on the functional form of consumer beliefs is difficult to reconcile with a supply model, in which firms condition their actions on consumer responses. 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. This assumption is always adopted by the theory literature and can be reconciled with a consistent supply model. Additionally, a recent empirical paper, Goettler and Gordon (2011), also adopted the same assumption as in this paper.

Information Set

Let $\mathbf{s}_{mt}$ denote the information set affecting agents' choices in 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. Consumers can also 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, $\epsilon_{mit} = (\{\epsilon_{ijt}\}_{j \in J_{mt}}, \epsilon_{im_0t})$.

Software Purchase

Let $G_{it}(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ denote type-i consumer's expected value function. Then, it can be written as

$$\begin{aligned} G_{it}(\mathbf{s}_{mt}, \mathbf{p}_{mt}) &= log\{\sum_{j \in J_{mt}} exp(x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt}) \\ &\quad + exp[\beta_c E G_{it+1}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt})]\}. \end{aligned} \tag{3}$$

[13] See Lee (2010), Gowrisankaran and Rysman (2011), Gowrisankaran, Park and Rysman (2011), and Hendel and Nevo (2007).

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

$$b_{ijt}(\mathbf{s}_{mt}, \mathbf{p}_{mt}) = \frac{exp(x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt})}{exp[\beta_c EG_{it+1}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt})] + \sum\limits_{j \in J_{mt}} exp(x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt})} \quad (4)$$

The demand for software $j \in J_{mt}$ at time t is

$$q_{jt}(\mathbf{s}_{mt}, \mathbf{p}_{mt}) = \sum_i n_{mit} b_{ijt}(\mathbf{s}_{mt}, \mathbf{p}_{mt}),$$

where n_{mit} is the number of active type-i consumers in submarket m at time t. Recall that each consumer is assumed to buy at most one software product in a submarket. Under this assumption, a consumer is no longer an active consumer in a submarket once she has made a purchase in that submarket. Meanwhile, new consumers enter a submarket once they purchase the compatible hardware. Therefore, the evolution of $\{n_{mit}\}$ follows

$$n_{mit+1} = n_{mit}(1 - \sum_{j \in J_{mt}} b_{ijt}) + n^e_{mit}, \quad (5)$$

where $n_{mit}(1 - \sum\limits_{j \in J_{mt}} b_{ijt})$ is the mass of consumers who do not buy in period t and remain active the next period; and $n^e_{mit} = N_{it} B_{ilt}$ is the mass of new consumers who purchase the compatible hardware l, as described in the previous subsection. Notice that the mass of consumers remaining in a submarket is endogenous to the historic entry and pricing behavior of all software firms in that submarket. The dynamics of entry and pricing introduce a dynamic evolution of the consumer distribution in the software submarket m.

Total Software Utility

In the previous subsection, I specify that the consumption value of a hardware product 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 able to purchase the affiliated software products. Recall that a consumer who purchases a hardware product starts to buy the affiliated software in the next period. Hence, Γ_{ilt} is type-i consumer's discounted total value being active in all submarkets affiliated with hardware l at time $t+1$,

$$\Gamma_{ilt} = \beta_c E\left[\sum_{m \in M_l} G_{it+1}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1})\right], \tag{6}$$

where M_l is the set of software submarkets affiliated with hardware l.

3.3 Software Pricing and Entry

In the 5th-generation U.S. video game market, a single software product was tiny compared to the whole market.[14] Hence, I assume that no single software firm can strategically influence the sales of hardware, and so that software firms do not take that effect into account when they make their choices. Under this assumption, strategic interactions occur only among software firms in the same submarket. Dube, Hitsch, and Chintagunta (2010) adopts the same assumption. Notice that this assumption would be more tenuous for more recent generations now that blockbuster games have become more common.

Below, I 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 An Incumbent Software Firm's Problem

Let c_l denote the unit cost of software affiliated to hardware l, including the production cost and the royalty fee paid to hardware provider l. Both of the two costs are time-invariant and platform-specific. An incumbent software firm's one-period profit depends on its own price choice this period (p_{jt}) and its competitors' prices ($\mathbf{p}_{-jt}$); moreover, it also depends on the state vector $\mathbf{s}_{mt}$ in the submarket m including the set of available products, product

[14] In this generation, the blockbuster games were smaller in magnitude. Among all Nintendo games, only three games took over 4% of the total game sales on the N64 platform, and only 21 games captured over 1% of the total game sales. Among all PS games, only five games captured over 1% of total game sales on the PS platform, none of them taking over 2%.

characteristics, and the consumer distribution.

Let β_f denote a software firm's discount factor. An incumbent's optimization problem is to pick a price to maximize its own discounted profit,

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

where $\pi_{jt}(\mathbf{s}_{mt}, p_{jt}, \mathbf{p}_{-jt}) = (p_{jt} - c_l)q_j(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ is firm j's one-period profit, the first expectation is taken with respect to competitors' price choices in this time period, and the second expectation is taken with respect to the distribution of future state variables and competitors' price choices in the future periods.

3.3.2 A Potential Software Entrant's Problem

Every period, there is finite number of potential entrants outside the software submarket m. Let E_{mt} denote the set of potential entrants. The entry cost of a potential entrant j is assumed to be $\lambda_l + \nu_{jt}$ where λ_l is the component that is common to all software affiliated with platform l 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. $F_\nu(\cdot)$.

Each potential entrant $j \in E_{mt}$ first draws an entry cost from a known distribution and then decides whether to enter. 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.

Let $y_{jt+1} = 1$ denote that entrant j decides to enter at time t. A potential entrant j's optimization problem is to compare the entry cost and the expected profit. The optimal strategy is to enter if the expected profit exceeds the entry cost and not to enter otherwise.

3.3.3 Perceived Strategy Function

Because a potential entrant's entry decision depends on its own entry cost shock ν_{jt} which is unobservable to consumers and other software 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

$$\rho_{jt}(\mathbf{s}_{mt}) = \int I(y_{jt+1}(\mathbf{s}_{mt}, \nu_{jt}) = 1) dF_{\nu}(\nu_{jt}),$$

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 software firms. The game has a Markov structure, and I assume that each software firm plays Markov strategies. In particular, if $\mathbf{s}_{mt} = \mathbf{s}_{m't}$, then firm j's decision in submarket m and m' are the same. Let $\Psi = \{\Psi_{jt}(\mathbf{s}_{mt})\}$ be a set of strategy functions or decision rules, one for each software firm, with $\Psi_{jt}(\mathbf{s}_{mt}) = p_{jt}(\mathbf{s}_{mt})$ if j is an incumbent firm and $\Psi_{jt}(\mathbf{s}_{mt}) = \rho_{jt}(\mathbf{s}_{mt})$ if j is a potential entrant.

3.3.4 Incumbent's Bellman Equation

Let $V_{jt}(\mathbf{s}_{mt} \mid \Psi)$ 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 software firms follow their strategies in Ψ. By Bellman's principle of optimality, it can be written as

$$V_{jt}(\mathbf{s}_{mt} \mid \Psi) = \max_{\tilde{p}_{jt}} \pi_{jt}(\mathbf{s}_{mt}, \tilde{p}_{jt}, p_{-jt}) + \beta_f E\left[V_{jt+1}(\mathbf{s}_{mt+1} \mid \Psi) \mid \mathbf{s}_{mt}, \tilde{p}_{jt}, \Psi_{-jt}\right], \qquad (7)$$

where

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

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

$$f_j(\mathbf{y}_{mt+1} \mid \mathbf{s}_{mt}, p_{jt}, \Psi_{-j}) = \prod_{k \in E_{mt}} \rho_{kt}(\mathbf{s}_{mt})^{y_{kt+1}}(1 - \rho_{kt}(\mathbf{s}_{mt}))^{1-y_{kt+1}}. \tag{8}$$

The optimal pricing strategy in response to profile Ψ is the solution of the right-hand side of equation (7), denoted as $p_{jt}(\mathbf{s}_{mt} \mid \Psi)$.

3.3.5 Entrant's Bellman Equation

Let $V^e_{jt}(\mathbf{s}_{mt}, \nu_{jt} \mid \Psi)$ 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 software firms behave according to strategy profile Ψ:

$$V^e_{jt}(\mathbf{s}_{mt}, \nu_{jt} \mid \Psi) = \underset{\tilde{y}_{jt+1}}{max}\ \tilde{y}_{jt+1}\left\{-\lambda_l - \nu_{jt} + \beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \Psi) \mid \mathbf{s}_{mt}, \Psi]\right\},$$

where

$$E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \Psi) \mid \mathbf{s}_{mt}, \Psi] = \int_{\xi_{mt+1}} \left[\sum_{\mathbf{y}_{mt+1}} V_{jt+1}(\mathbf{s}_{mt+1} \mid \Psi) f_j(\mathbf{y}_{mt+1} \mid \mathbf{s}_{mt}, \Psi)\right] d\xi_{mt+1}$$

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

$$f_j(\mathbf{y}_{mt+1} \mid \mathbf{s}_{mt}, \Psi) = \prod_{k \in E_{mt}, k \neq j} \rho_{kt}(\mathbf{s}_{mt})^{y_{kt+1}}(1 - \rho_{kt}(\mathbf{s}_{mt}))^{1-y_{kt+1}}, \tag{9}$$

where the jth dimension of $\mathbf{y}_{mt+1}$ is equal to one.

The optimal entry decision follows a cutoff rule characterized by

$$y_{jt+1}(\mathbf{s}_{mt}, \nu_{jt} \mid \Psi) = \begin{cases} 1, & if\ \nu_{jt} \leq \bar{\nu}_{jt}(\mathbf{s}_{mt} \mid \Psi) \\ 0, & otherwise \end{cases}$$

where

$$\bar{\nu}_{jt}(\mathbf{s}_{mt} \mid \Psi) = \beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \Psi) \mid \mathbf{s}_{mt}, \Psi] - \lambda_m$$

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

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

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

$$\begin{aligned} V^e_{jt}(\mathbf{s}_{mt} \mid \Psi) \;\; &= \;\; \underset{\tilde{\rho}_{jt}}{max} \; - \int_{\nu_{jt} < F_\nu^{-1}(\tilde{\rho}_{jt})} \nu_{jt} dF_\nu(\nu_{jt}) \\ &\quad + \tilde{\rho}_{jt} \left\{ -\lambda_l + \beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \Psi) \mid \mathbf{s}_{mt}, \Psi] \right\}. \end{aligned} \tag{10}$$

3.4 Equilibrium Concept

This paper adopts the Markov Perfect Equilibrium (MPE) concept. The MPE in this model is defined by a set of value functions, $\{G_{it}(\mathbf{s}_{mt}, \mathbf{p}^*_{mt}), H_{it}(\mathbf{S}_t)\}_{i=1}^I$ and $\{V_{jt}(\mathbf{s}_{mt})\}_{j \in J_{mt}}$, a set of price functions, $\{p^*_{jt}(\mathbf{s}_{mt})\}_{j \in J_{mt}}$, and a set of entry functions, $\{\rho^*_{jt}(\mathbf{s}_{mt})\}_{j \in E_{mt}}$, such that equation (1) - (10) are simultaneously satisfied at every state $\mathbf{s}_{mt}$. In other words, the equilibrium is the fixed point of the game defined by equations (1) - (10), with the following properties.

(i) Software Firms. Equation (7) implies that in equilibrium, when faced with state $\mathbf{s}_{mt}$, each incumbent software firm's pricing policy is a best response to other software firms' strategies and consumers' behavior at that state. Meanwhile, equation (10) implies that in equilibrium, when faced with state $\mathbf{s}_{mt}$, each entrant's entry policy is a best response to other software firms' strategies and consumers' behavior at that state.

(ii) Consumers. Equation (3) and (4) imply that when faced with a state $\mathbf{s}_{mt}$ and price $p(\mathbf{s}_{mt})$, consumers who own a hardware rationally anticipate software firms' future pricing and entry, and optimally make purchase decisions of software. At the same time, equation (1) and (2) imply that in equilibrium, consumers who do not own any hardware make purchase decisions of hardware by maximizing inter-temporal utility. In addition, the value of a hardware is given by the equation (6).

(iii) State Transition. Software firms take into account the effect of their actions on the evolution of states in the submarket. The transition of consumer distribution follows the equation (5). In the eyes of incumbent software firms, the transition of the product availability follows the equation (8); and in the eyes of potential entrants who decide to enter, it follows the equation (9).

4 Bayesian Estimation

In this section, I describe the estimation procedure 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 prices and quantity sold of each hardware product in each time period; and (ii) the availability, characteristics, prices, and quantity sold of each software in each time period across M independent software submarkets. Hence, $data = \{\mathbf{P}_t, \mathbf{Q}_t, \{\mathbf{y}_{mt}, \mathbf{x}_{mt}, \mathbf{p}_{mt}, \mathbf{q}_{mt}\}_{m=1}^{M}\}_{t=1}^{T_d}$, 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

Let $\mathcal{L}(data \mid \theta)$ denote the likelihood. Rather than using the maximum likelihood estimation method, I employ the Bayesian MCMC method to sample the parameter vector θ from its posterior distribution,

$$\mathcal{P}(\theta \mid data) \propto \mathcal{L}(data \mid \theta)\pi(\theta), \tag{11}$$

where $\pi(\theta)$ is the prior distribution of the parameter vector θ.

4.2 Likelihood Contributions

The demand for hardware is a dynamic discrete choice model. I assume that the unobserved (to researcher) platform-specific demand shifters ζ_{lt} are normally distributed with mean zero and variance σ_ζ^2, independent across all products and over time. The distribution of the aggregate demand shocks generate the distribution of the units sold of each hardware in each time period. Conditional on the state $\mathbf{S}_t$, the joint density of the sales of all hardware at time t is

$$\mathcal{L}_Q(\mathbf{Q}_t \mid \mathbf{S}_t; \theta) = \prod_l [\phi(\zeta_{lt}/\sigma_\zeta)/\sigma_\zeta] \mid \left(J_{(\mathbf{Q}_t \to \zeta_{lt})}\right)^{-1} \mid . \tag{12}$$

where $\phi(\cdot)$ is the *pdf* of the standard normal distribution and $J_{(\mathbf{Q}_t \to \zeta_{lt})}$ is the Jacobian matrix.

To specify the likelihood contribution of the demand for software, I assume that the unobserved game-specific demand shifters ξ_{jt} are normally distributed with mean zero and variance σ_ξ^2, independent across all products and over time.[15] The distribution of the aggregate demand shocks generate the distribution of the units sold of each existing software product in each time period. Conditional on the state $(\mathbf{s}_{mt}, \mathbf{p}_{mt})$, the joint density of the sales of all existing software products in submarket m at time t is

$$\mathcal{L}_q(\mathbf{q}_{mt} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt}; \theta) = \prod_{j \in J_{mt}} [\phi(\xi_{jt}/\sigma_\xi)/\sigma_\xi] \mid \left(J_{(\mathbf{q}_{mt} \to \xi_{mt})}\right)^{-1} \mid, \tag{13}$$

[15] In 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.

To evaluate the likelihood, I need to derive ξ_{jt}, which is described in the next subsection, and evaluate the Jacobian, $J_{(\mathbf{q}_{mt}\rightarrow\xi_{mt})}$, which is derived in Appendix B.1.

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. I 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.[16] Furthermore, it is assumed to follow a log-normal distribution with mean zero and variance σ^2_{ς}, independent over time and across products. Hence, conditional on the state vector $\mathbf{s}_{mt}$, the likelihood contribution of incumbent $j \in J_{mt}$ at time t is given by

$$\mathcal{L}_p(p_{jt} \mid \mathbf{s}_{mt};\theta) = \frac{1}{\sigma_{\varsigma}}\phi\left(\frac{ln[\tilde{p}_{jt}/p_j(\mathbf{s}_{mt},\theta)]}{\sigma_{\varsigma}}\right). \tag{14}$$

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^2_{ν}.[17] Hence, conditional on the state vector $\mathbf{s}_{mt}$, the likelihood contribution of entrant $j \in E_{mt}$ is

$$\begin{aligned}\mathcal{L}_y(y_{jt+1} \mid \mathbf{s}_{mt};\theta) \quad &= \quad \left(\Phi\left[\frac{\beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \mathbf{s}_{mt};\theta)] - \lambda}{\sigma_{\nu}}\right]\right)^{y_{jt+1}} \\ &\times \left(1 - \Phi\left[\frac{\beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \mathbf{s}_{mt};\theta)] - \lambda}{\sigma_{\nu}}\right]\right)^{1-y_{jt+1}}. \end{aligned}\tag{15}$$

Therefore, the likelihood can be written as

$$\mathcal{L}(data \mid \theta) = \prod_{t=1}^{T_d}\left\{\mathcal{L}_Q(\mathbf{Q}_t \mid \mathbf{S}_t;\theta)\prod_{m=1}^{M}\left[\mathcal{L}_q(\mathbf{q}_{mt} \mid \mathbf{s}_{mt},\mathbf{p}_{mt};\theta)\prod_{j\in J_{mt}}\mathcal{L}_p(p_{jt} \mid \mathbf{s}_{mt};\theta)\prod_{j\in E_{mt}}\mathcal{L}_y(y_{jt+1} \mid \mathbf{s}_{mt};\theta)\right]\right\}.$$

[16] In the data set, 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}.

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

4.3 Estimation Algorithm

The estimation procedure involves two loops: in the outer-loop, I use the Metropolis-Hastings algorithm to update the structural parameters; and in the inner-loop, for a given parameter vector, I update each agent's value function and best response function by using the results from previous MCMC iterations. This estimation procedure does not fully solve the dynamic model but incorporates the approximation approach. Hence, it significantly alleviates the computational burden and makes Bayesian estimation method applicable to dynamic games. Below, I describe the estimation procedure in detail.

4.3.1 Outer-Loop: Metropolis-Hastings (MH) Step

The posterior distribution in equation (11) 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 (see McCulloch and Rossi 1994, Albert and Chib 1993, Imai, Jain and Ching 2007). Based on the model, I partition all parameters into four blocks: (i) the first block includes all parameters directly affecting consumer purchase decisions of hardware, i.e., the parameters in the utility function of hardware, $\theta_1 = (\gamma, \alpha_i, \sigma_\zeta)$; (ii) the second block includes all parameters directly affecting consumer purchase decisions of software, i.e., the parameters in the utility function of software, $\theta_2 = (\psi, \varphi_i, \sigma_\xi)$; (iii) the third block includes all parameters directly affecting incumbent software firms' pricing decisions, i.e., the unit cost of games sold on each platform and the standard deviation of the pricing error, $\theta_3 = (c_{Saturn}, c_{PS}, c_{N64}, \sigma_\varsigma)$; and (iv) the last block includes all parameters directly affecting entrants' entry decisions, i.e., the mean and the standard deviation of game producers' entry cost to each platform, $\theta_4 = (\lambda_{Saturn}, \lambda_{PS}, \lambda_{N64}, \sigma_\nu)$.

Consider a particular iteration k. For each block l, the procedure goes as follows.

The first step is to draw the candidate parameter vector $\theta_l^{*(k)}$ from a proposed density.

As usual in the literature,[18] 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, given by

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

where $\mathcal{L}_l(\cdot \mid \theta)$ equals to equation (12), (13), (14) and (15), 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.

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

4.3.2 Inner-Loop: Fixed Point (FP) Step

Evaluating the acceptance-rejection ratio in the outer-loop requires evaluating the likelihood which requires solving the dynamic game given a vector of parameters. The computation difficulty comes in two parts. One part is computing the equilibrium strategies of all agents which are the fixed points of the best response system. The other part is computing each agent's value function given other agents play their equilibrium strategies, which is the fixed point of a single-agent dynamic programming (DP) problem. In this paper, I develop a new procedure of solving the fixed point of a dynamic model suitable for use in conjunction with the Bayesian MCMC estimation.

For a given draw of the parameter vector along the MCMC chain, I first randomly pick a subset from the entire state space for each period; then, for each given point in the subset, I non-parametrically approximate each agent's equilibrium strategy and value function by using the pseudo-best response functions and pseudo-value functions from previous MCMC iterations; after that, I adopt an interpolation approach to obtain each agent's continuation

[18] See Jiang, Manchanda and Rossi (2009), and Imai, Jain, and Ching (2009).

value, solve for each agent's best response function (pseudo-best response function) and value function (pseudo-value function) given that other agents play their equilibrium strategies, and store these pseudo-best response functions and pseudo-value functions for future MCMC iterations. This procedure is similar to the method of Pakes and McGuire (2001). In their algorithm, the continuation value is approximated by the average of the returns from past outcomes of the algorithm, and the value and policy functions are updated at a recurrent class of points (rather than at all possible points) in the state space.

Nonparametric Approximation of Equilibrium Strategy

One 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.[19] 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 easily deal with the unobservables.[20]

In this paper, I propose to use a kernel method to approximate the equilibrium strategies using the pseudo-best response of the past iterations in which the parameter vector is "close" to the current parameter vector. The equilibrium strategy of software producer j in iteration k is computed as

$$\hat{\Psi}_{jt}^{(k)}(\mathbf{s}_{mt}, \theta) \quad = \quad \sum_{n=1}^{N(k)} \Psi_{jt}^{(k-n)}(\mathbf{s}_{mt}, \theta^{*(k-n)}) \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})}, \tag{16}$$

[19]The general idea is to start with an initial guess at the value function and substitute that into the right-hand side 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, 2001).

[20]Hu and Shum (2011) consider nonparametric identification of dynamic models with general unobservables.

where $\Psi_{jt}^{(k)}$ 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's optimization problem, $p_{jt}^{(k)}(\mathbf{s}_{mt}, \theta)$, and Appendix B.2 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

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

In essence, the equilibrium strategies are approximated by the weighted average of pseudo-best response of past 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, as the number of MCMC iterations and the number of past iterations for approximating the 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 functions converge to the true posterior distributions.

Basically, I combine the MH step with the FP step for a single iteration. I do not fully solve for the equilibrium of the dynamic model but use the results from previous MCMC iterations. This spirit is similar to Aguirregabiria and Mira (2007)'s nested pseudo likelihood method that gradually updates the equilibrium probabilities and recursively obtains the estimators.

Non-Parametric Approximation of Value Function

To compute the value function at a given state point, the conventional estimation methods iterate the Bellman operator until convergence. It is computationally difficult for relatively complicated models. The IJC proposes a nonparametric 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 IJC 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. Following the same idea as in Ishihara and Ching (2011), 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 value function at state $(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ in iteration k is approximated as

$$\hat{G}_{it}^{(k)}(\mathbf{s}_{mt}, \mathbf{p}_{mt}, \theta) = \sum_{n=1}^{N(k)} G_{it}^{(k-n)}(\mathbf{s}_{mt}, \mathbf{p}_{mt}, \theta^{*(k-n)}) \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})}, \tag{17}$$

where $K_h(\cdot)$ is a multivariate kernel with bandwidth $h > 0$, and $G_{it}^{(k)}(\mathbf{s}_{mt}, \mathbf{p}_{mt}, \theta)$ is consumer's pseudo-value function at state $(\mathbf{s}_{mt}, \mathbf{p}_{mt})$ conditional on that all software firms playing the equilibrium $\hat{\Psi}^{(k)}$

$$\begin{aligned} G_{it}^{(k)}(\mathbf{s}_{mt}, \mathbf{p}_{mt}, \theta) &= ln\{\sum_{j \in J_{mt}} exp(x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt}) \\ &\quad + exp\left(\beta_c \hat{E}[G_{it+1}^{(k)}(\mathbf{s}_{mt+1}, \hat{\mathbf{p}}_{mt+1}^{(k)}, \theta) \mid \mathbf{s}_{mt}, \mathbf{p}_{mt}, \hat{\Psi}^{(k)}, \theta]\right)\} \end{aligned} \tag{18}$$

The approximated value function given by equation (17) 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 value functions increase, the pseudo-value function converges to the true value functions, and the posterior parameter draws based on the pseudo-value functions converges to the true posterior distributions. Moreover, the convergence of the approximated value function to the true value function requires that $N(k) \to \infty$ and $k - N(k) \to \infty$ as $k \to \infty$.

A similar method applies to computing the software firm's value function

$$\hat{V}_{jt}^{(k)}(\mathbf{s}_{mt},\theta) \quad = \quad \sum_{n=1}^{N(k)} V_{jt}^{(k-n)}(\mathbf{s}_{mt},\theta^{*(k-n)}) \times \frac{K_h(\theta-\theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta-\theta^{*(k-n)})} \tag{19}$$

where $V_{jt}^{(k)}(\mathbf{s}_{mt},\theta)$ is the incumbent software firm j' pseudo-value function at state $\mathbf{s}_{mt}$ conditional on that all other software firms playing the equilibrium $\hat{\Psi}^{(k)}$:

$$V_{jt}^{(k)}(\mathbf{s}_{mt},\theta) = \underset{\tilde{p}_{jt}}{max}\, \pi_{jt}(\mathbf{s}_{mt},\tilde{p}_{jt},\hat{p}_{-jt}^{(k)}) + \beta_f \hat{E}\left[V_{jt+1}^{(k)}(\mathbf{s}_{mt+1},\theta) \mid \mathbf{s}_{mt},\tilde{p}_{rjt},\hat{\Psi}^{(k)},\theta\right]. \tag{20}$$

Store the solved best response functions (pseudo-best response functions), $p_{jt}^{(k)}(\mathbf{s}_{mt},\theta)$, and the solved value functions (pseudo-value functions), $V_{jt}^{(k)}(\mathbf{s}_{mt},\theta)$ and $G_{it}^{(k)}(\mathbf{s}_{mt},\mathbf{p}_{mt},\theta)$, for future MCMC iterations.

Interpolation

However, to obtain the expected value functions in equation (18) and equation (20), we still need to compute equations (17) and (19) for every possible point of the state space. Due to the "curse of dimensionality",[21] it is computationally burdensome to achieve it even with the nonparametric approximation method proposed above.

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 procedure to deal with the large state space problem. In the first step, I randomly choose a subset of the state points

[21] The 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 $(KM)^N$.

every period, and obtain the values at those points with the non-parametric approximation approach described above. 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 approximated 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 a finite number of times and then take the average of the predicted values. This estimation procedure 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 procedure 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 non-linearly in optimization problems.[22]

Recall that the state vector in the model includes the availability and characteristics of each software product in a submarket, and the distribution of remaining consumers of each type. In addition, consumers can also observe the price of each software 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 software product availability depends on all potential entrants' entry choices up to the previous period; and the software price is chosen by incumbent software 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 software products available

[22]Note 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.

for purchase, the distribution of remaining consumers of each type, his/her own mean utility from the top-ranked software product, and the average of his/her mean utility from all existing products. Incumbent software firms trace the time periods to the end, the number of software 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 incumbents do. The only difference is that they trace the expected value of new product instead of the value of its own product.

Computing ζ_{lt} and ξ_{jt}

Once we obtain the consumer's continuation values, we can compute each consumer's probability of purchasing from equation (4) and then the predicted demand of each product. To obtain the likelihood contribution of demand in equation (12), 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 this paper 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, ζ_{lt}, given by

$$\zeta_{lt}^{(k)} = \zeta_{lt}^{(k-1)} + ln(\tilde{Q}_{lt}) - ln\left(Q_{lt}^{(k)}(\mathbf{S}_t, \theta)\right).$$

5 Estimation Results

5.1 Econometric Details

Consumer Heterogeneity. For simplicity but without loss of generality, I assume two consumer types who have different sensitivity to price: high-type consumers and low-type consumers.[23] At the outset, 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 hardware and software. In particular, consumers' purchase decision of software in a submarket determines the number of consumers remaining for the next period, and their purchase decision of hardware determines the number of new consumers who enter the software market next period. In this paper, I set N_0 to 100 million.

Discount Factors. Previous literature has noted that it is difficult to estimate discount factors, so I do not attempt to estimate the discount factors for consumers and software firms (β_c and β_f). Instead, I set the discount rates to 0.95, which is lower than the monthly interest rate. However, previous studies in experimental and behavioral economics have found that the discount factor is lower than the interest rate.

Prior Distribution. In order to estimate the model it is necessary to specify the prior distribution for the parameters to be estimated. Consumer preference to product characteristics (ψ and γ) and consumer sensitivity to price (φ_i and α_i) follow normal distributions with means of zero and large standard deviations. The initial share of high-type consumers (δ) follows a uniform distribution on the interval [0,1]. To guarantee that cost parameters and standard deviations are non-negative, their prior distributions are log-normal with means of zero and large standard deviations.

[23] The 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.

Initial Guess of Equilibrium Strategies and Value Functions. To compute the initial guess for consumer value functions and incumbent value functions I assume that both consumers and software firms are myopic. The initial guesses for product prices are the predicted values from a hedonic regression of price on state variables. I compute the initial guess of the entry probability based on the initial value functions.

5.2 Posterior Statistics

I draw 100,000 samples from the posterior distribution and use the last 50,000 samples to derive the posterior means and standard deviations. The last 50,000 samples are reported in Table 2. I also compute the posterior means and standard deviations from the last 25,000 samples. The findings in the two sets of samples are not statistically different. I repeat this procedure several times and find that the posterior statistics are robust to initial values of parameters that are drawn from their prior distributions. I conclude that the samples I use to compute the posterior statistics are drawn from a stable distribution.

The estimates in the consumer utility function of hardware are consistent with our expectation. High-type consumer price sensitivity to hardware is 0.018 and low-type consumer price sensitivity to hardware is 0.064. The numbers are positive because they enter the utility function as a negative term. Consumers obtain higher utility from purchasing consoles in November or December, probably because consoles are good gifts during the holiday season. High-type consumers correspond to 14.6 percent of the potential market at the beginning of the console lifecyle.

The estimates in the consumer utility function of software are consistent with our expectation. Nintendo 64 games generate the highest utility because the console is more technologically advanced than the other two. Consumers favor the games with high online ratings, and consumers dislike games that have been in the market for a long time, partly because most sports games are designed based on the latest tournaments. Consumers obtain higher utility from purchasing games in November or December, probably because they can spend more time playing games during holiday season. High-type consumer price sensitivity to software is 0.014 and low-type consumer price sensitivity to software is 0.051.

Table 2: Posterior Means and Standard Deviations

	Last 50,000 Samples		Last 25,000 Samples	
	Mean	Std dev	Mean	Std dev
Block 1: Demand for Hardware				
α_1 (H-type consumer price sensitivity)	0.018	0.004	0.018	0.003
α_2 (L-type consumer price sensitivity)	0.064	0.015	0.064	0.012
γ_{Nov} (Nov. dummy)	0.745	0.217	0.746	0.219
γ_{Dec} (Dec. dummy)	2.405	0.903	2.399	0.903
σ_ζ (std of hardware demand shocks)	0.105	0.581	0.106	0.612
δ (initial share of H-type consumers)	0.146	0.093	0.146	0.093
Block 2: Demand for Software				
ψ_{N64} (dummy for N64 games)	1.544	0.343	1.537	0.074
ψ_1 (online rating score of games)	0.169	0.090	0.171	0.068
ψ_2 (game age if new)	-0.333	0.097	-0.341	0.023
ψ_3 (game age if old)	-0.189	0.062	-0.193	0.033
ψ_{Nov} (Nov. dummy)	0.239	0.074	0.241	0.041
ψ_{Dec} (Dec. dummy)	0.672	0.206	0.675	0.188
φ_1 (H-type consumer price sensitivity)	0.014	0.005	0.014	0.002
φ_2 (L-type consumer price sensitivity)	0.051	0.023	0.052	0.017
σ_ξ (std of software demand shocks)	2.739	0.107	2.742	0.050
Block 3: Software Pricing				
c_{Saturn} (unit cost of games for Saturn)	14.652	2.176	14.774	2.322
c_{PS1} (unit cost of games for PS)	10.755	1.601	10.731	1.325
c_{N64} (unit cost of games for N64)	18.458	1.629	18.523	1.726
σ_ς (std of price error)	0.879	0.220	0.895	0.154
Block 4: Software Entry				
λ_{Saturn} (mean of entry cost to Saturn)	4.717	0.428	4.7292	0.170
λ_{PS1} (mean of entry cost to PS)	3.663	0.505	3.700	0.417
λ_{N64} (mean of entry cost to N64)	4.613	0.142	4.614	0.086
σ_ν (std of entry cost shocks)	2.487	0.119	2.493	0.105

The cost per unit is \$14.7, \$10.8 and \$18.5 for games released for Saturn, PS and N64, respectively. As Coughlan (2001) reported, the production and packaging cost for 32-bit CD-ROM games is around \$1.5 per disc. Therefore, the royalty fees charged by Sega and Sony were around \$13.2 and \$9.3 per copy sold. The unit cost of N64 games is much higher than Saturn games and PS games because Nintendo used ROM cartridges to store games, so their production expense was much higher than the production expense for the compact disc format that competitors used.

The average entry cost of Saturn games, PS games and N64 games are \$4.7 million, \$3.7 million and \$4.6 million, respectively. The standard deviation of entry cost is \$2.5 million. Saturn games' research and development cost was on average significantly higher than PS games, partly due to Saturn's dual-CPU architecture and more complex graphics, even though both Sega and Sony adopted very similar technology.[24]

5.3 Numerical Solution

To examine the fit of the model, I treat the posterior means of the last 50,000 samples as the estimated values of the parameters and numerically solve the model by using the approximation method described in the previous section. The algorithm is programmed in Fortran 95 and converged smoothly for those parameters. The convergence of the numerical solution indicates the existence of a unique equilibrium at those parameter values.[25] However, it is possible that multiple equilibria exist for other parameter values.

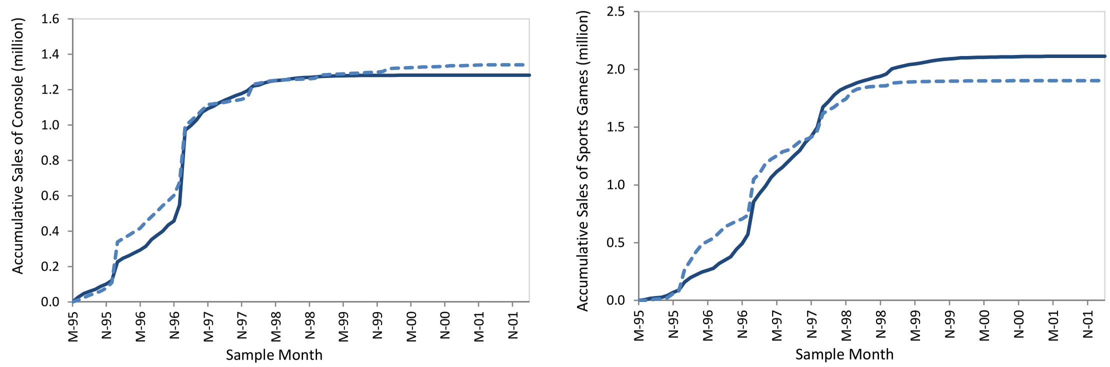
I now compare the predicted values to those observed in the data. Figure 6 (a) compares the predicted and the observed number of console owners. Figure 6 (b) compares the predicted and the observed cumulative sales of sports games. Overall, the model fits the data very well.

[24] See http://en.wikipedia.org/wiki/Sega_Saturn#cite_note-16. "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 Virtual Fighter development.

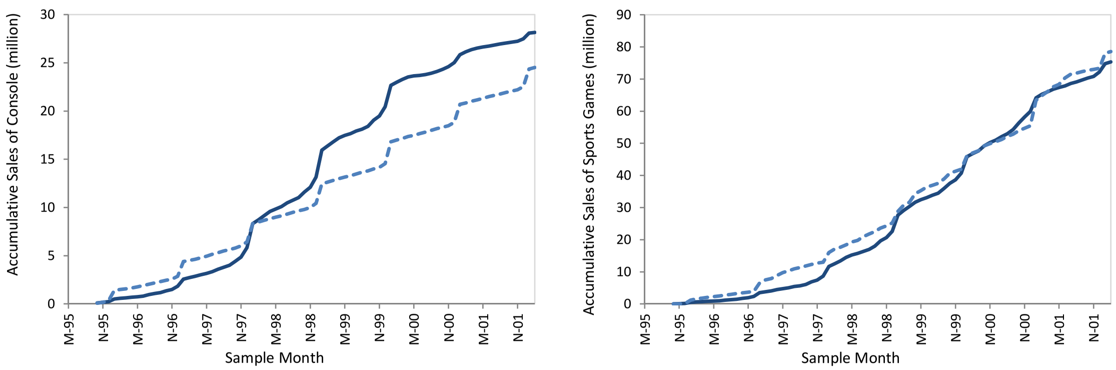
[25] Generally speaking, it is difficult to analytically prove the existence and uniqueness of a MPE in pure strategy for dynamic oligopoly models. I have proved that, under some restrictions, there exists a unique equilibrium in pure strategy for a dynamic oligopoly pricing model with forward-looking consumers. Yet it is extremely hard to go further to show the equilibrium existence for this model which also contains software firms' dynamic entry and consumers' self-selection to platforms. Without analytical solutions, I am unable to formally state whether an equilibrium exists and whether it is unique.

Figure 6: Actual vs. Fitted Accumulative Sales of Consoles and Games

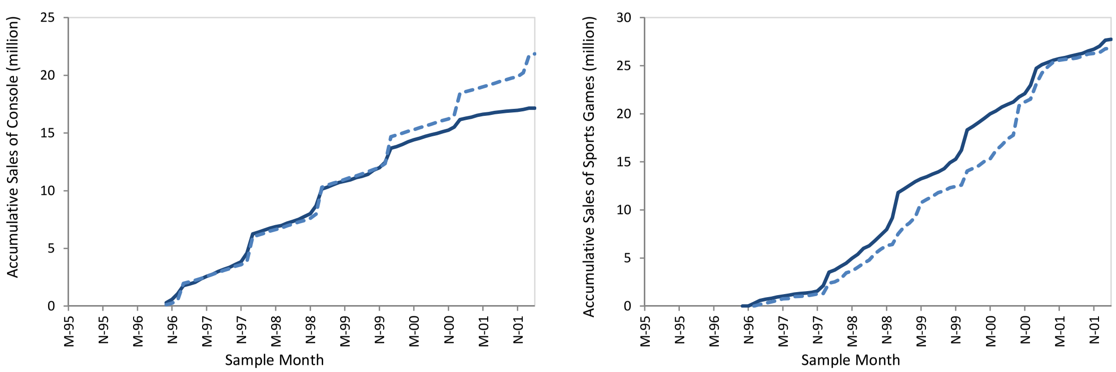
(a) Sega Saturn



(b) Sony PlayStation



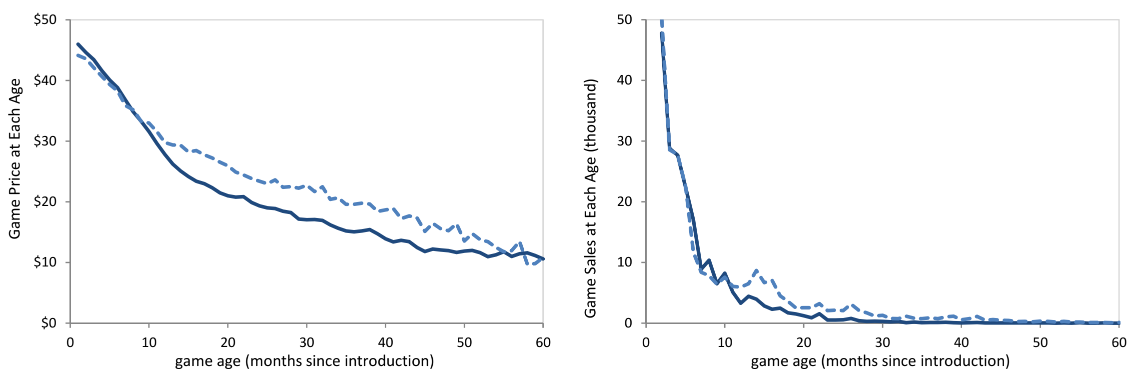
(c) Nintendo 64



Note: Solid lines represent actual data. Dashed lines are fitted values.

Figure 7 (a) compares the predicted and the observed prices across all sports games at each game's age, or months since the game's introduction. The figure indicates that the proposed model is able to explain the declining pattern of game price. However, the predicted price does not drop as quickly the observed price. One possible reason is that the model does not consider the second-hand market that contributes to the declining game price in the data. Figure 7 (b) compares the predicted and the observed unit sales across all sports games at each age. The figure shows that the model fits the data very well.

Figure 7: Actual vs. Fitted Game Price and Unit Sales in Age



Note: Solid lines represent actual data. Dashed lines are fitted values.

6 Counterfactual Simulations

In this section, I make use of the recovered parameters in the demand and supply model to conduct counterfactual exercises. The goal is to explore what contributed to the Sega Saturn's failure. To be more specific, would Sega have survived if it had taken alternative options such as charging lower console prices or subsidizing software entry in its initial stage?

In the first counterfactual exercise, I examine what would happen if Sega had lowered its console prices by $100 for the first two years. The second counterfactual exercise looks at whether Sega would have survived if it had reduced game producers' entry cost. In each scenario, I change Sega's entry fee and simulate both the number of console owners and the number of associated games. This approach might be doubted because the competing platforms would react to the change. Unfortunately, this paper does not focus on the pricing competition among platforms. As a result, I do not know competing platforms' best response functions. However, the simulation results of changing one platform and fixing the others still shed some light on how a platform's prices affect that platform's formation process. The simulation quantitatively measures how responsive consumers and software firms are to a platform's introductory prices.

6.1 Lowering Console Price

Figure 2 (a) presents the prices of the three fifth-generation consoles in every month during the sample period. Console prices were generally declining over time. The figure also shows that Sega's console prices were $100 higher than competitors' prices at the same age for the first two years. A high console price discouraged consumer entry and hence software entry. It is speculated that this contributed to Sega's failure. In this counterfactual, I consider what would happen if Sega had reduced its console prices by $100 for the first two years. The goal is to investigate whether Sega could have survived if it had charged lower prices to consumers. The simulated results are reported in Table 3.

Table 3: Results of Counterfactural 1: Lowering Sega's Console Price

	Observed Data	Counterfactual
Sega Saturn:		
Hardware owners (2/2002, m)	1.28	9.71
Software accumulative units sold (2/2002, m)	2.11	16.02
Sony PlayStation:		
Hardware owners (2/2002, m)	28.25	20.95
Software accumulative units sold (2/2002, m)	75.97	59.60
Nintendo 64:		
Hardware owners (2/2002, m)	17.17	14.91
Software accumulative units sold (2/2002, m)	27.84	21.65

With the proposed price schedule for Sega, the number of Sega console owners would increase by around 8 million at the end of the sample period (February 2002), and its cumulative sports game sales would increase by around 14 million units. Because the $100 price reduction is not a negligible number, the impact on consumer adoption of hardware and software sales are considerably large. In one-sided markets, the responsiveness of the quantity demanded to a price change depends on consumers' elasticity. However, in two-sided markets, the responsiveness of the quantity demanded not only depends on consumers' elasticity on the good but on the strength of the indirect network effects. For example, a lower console price attracts new users and hence new games; games attract extra new users, and so on. When consumers are very sensitive to hardware price and game producers highly value the installed base of users, a change in the hardware price would have a huge impact. However, when consumers are inelastic and game producers do not value the installed base as much, the impact of a change in the hardware price would not be that big.

Due to the lower console price, Sega would lose $421.5 million on the console side of the market. However, the increased sales of the sports games would lead to an extra $183.61 million in royalty fees from the game producers. Since I only know the increased royalty fees from the sports games, the total net gain from this subsidy package is unknown. If non-sports games behave the same way as the sports games, then the increased royalty fees from all games would be scaled up to 831.44 (= 183.61*240/53) million dollars[26]. As a result, the total net gain would be $409.94 million. Therefore, reducing the console price in the initial stage would be very effective in promoting Sega's platform size and increasing its profit.

Sony PlayStation would suffer from Sega's lower console price strategy. PlayStation's installed base of users would decrease by around 7 million, and its accumulated sports game sales would decrease by around 16 million. Nintendo 64 would not be affected as much. PlayStation technology and entry time were almost the same as Saturn's, and therefore was the main rival of Saturn. However, N64 was more technologically advanced and entered the market one and a half years later than Saturn, and hence was not Saturn's direct competitor.

[26]The sample contains 240 Saturn games in total among which 53 games are sports games.

6.2 Subsidize Software Entry

The estimates of the entry costs of the game producers to the three platforms are considerably different from each other. Saturn's entry cost for games is on average significantly higher than PS games. High entry cost discourages software entry and thus discourages consumer entry. In this counterfactual, I consider what would happen if Sega had given game producers a subsidy of $1 million per game title. Even though a software producer's entry cost does not directly go to platform providers' pockets, platforms can still strategically influence these software entry fees, either by offering easier-learning development technology or by directly subsidizing software research and development.

Table 4: Results of Counterfactural 2: Lowering the Entry Cost of Sega Games

	Observed Data	Counterfactual
Sega Saturn:		
Hardware owners (2/2002, m)	1.28	4.73
Software accumulative units sold (2/2002, m)	2.11	7.36
Sony PlayStation:		
Hardware owners (2/2002, m)	28.25	24.78
Software accumulative units sold (2/2002, m)	75.97	68.34
Nintendo 64:		
Hardware owners (2/2002, m)	17.17	16.08
Software accumulative units sold (2/2002, m)	27.84	25.22

Table 4 reports the simulation results. Sega would attract 12 new sports games. This requires Sega pay a $65-million subsidy to the producers of sports games. Sega's hardware users would increase by around 3.46 million at the end of the sample period, and its cumulative sports game sales would increase by around 5.25 million. The increased sales of sports games would generate $69.3 million in royalty fees for Sega. As a result, this subsidizing package would generate a net gain of $4.3 million from the sports games. If non-sports games behave the same as sports games, then the total net gain would be scaled up to 19.47 (= 4.3*240/53) million dollars. Therefore, subsidizing software entry would not help Sega very much. This result indicates that when compared to the high console prices to consumers, the high entry cost to software firms was not the main cause of Sega's failure. This might be because software firms were not as sensitive to entry costs, or because consumers did not value software variety as much, or both.

Under this counterfactual, PlayStation's console owners would decrease by around 3.47 million and its cumulative sports game sales would drop by around 7.63 million. Nintendo 64 would not suffer very much.

7 Concluding Remarks

In this paper, I present a dynamic structural model to study consumers' demand for hardware and associated software, and software firms' entry and pricing decisions. I examine Sega's alternative strategic options such as reducing its console prices for the first two years or subsidizing software entry. For each alternative option, I simulate the number of participants and Sega's revenue. The results show that Sega would have survived if it had reduced console price by $100 in the launching stage, and that subsidizing software entry would not have helped Sega walk out of the death spiral.

This paper provides a framework to structurally model a two-sided market, especially a hardware-software market. On the consumer side, the value of a hardware platform depends on the expected value of optimally purchasing software products associated with that platform. Moreover, the number of potential buyers for a software product depends on the number of users who have adopted the compatible hardware. The demand system of the model accounts for dynamic selection of forward-looking and heterogeneous consumers into platforms for affiliated software products, and allows for the contingency of platforms' value on the availability and prices of the affiliated products. By incorporating the complementarity between hardware and software, I am able to examine the effects that subsidizing entry of the software side has on consumer entry to a platform.

On the software side, software producers compete within a submarket. Incumbents choose their prices, and potential entrants choose whether to enter. Potential entrants strategically account for competitor reactions and consumer responses. I have investigated the impacts that subsidizing entry on the consumer side has on software producers' entry and price choices. Counterfactual experiments demonstrate that lowering console price has significantly increased consumer entry and therefore software entry.

This paper also develops a practical Bayesian MCMC procedure for structural estimation of dynamic models. I use the outcomes from past MCMC iteration to approximate each agent's equilibrium strategy and value function for the current draw of parameter vector. This estimation procedure significantly reduces computational burden. To avoid computing the value function at all possible points of the state space, I combine the nonparametric approximation method and interpolation method. I also implement the estimation procedure to estimate a dynamic model in the video game market. The estimation procedure can be used to estimate other dynamic models, especially those with unobserved heterogeneity and large state space.

References

[1] Aguirregabiria, V. and Mira, P. (2007): "Sequential Estimation of Dynamic Discrete Games", *Econometrica,* 75(1), 1-53.

[2] Albert, H. and Chib, S. (1993): "Bayesian Analysis of Binary and Polychotomous Response Data", *Journal of the American Statistical Association*, 88, 669-675.

[3] Armstrong M. (2006): "Competition in Two-Sided Markets", *RAND Journal of Economics*, 37(3), 668-691.

[4] Bajari, P., Benkard, L. and Levin, J. (2007): "Estimating Dynamic Models of Imperfect Competition", *Econometrica,* 75(5), 1331-1370.

[5] Caillaud, B. and Jullien, B. (2003): "Chicken & Egg: Competition among Intermediation Service Providers", *The RAND Journal of Economics*, 34(2): 309-328.

[6] Ching, A., Imai, S., Ishihara, M., and Jain, N. (2009): "A Practioner's Guide to Bayesian Estimation of Discrete Choice Dynamic Programming Models", working paper.

[7] Doraszelski, U. and Satterthwaite, M. (2010): "Computable Markov-Perfect Industry Dynamics", *RAND Journal of Economics*, 41(2), 215-243.

[8] Ericson, R. and Pakes, A. (1995): "Markov-Perfect Industry Dynamics: A Framework for Empirical Work", *Review of Economic Studies,* 62, 53-82.

[9] Evans, D. and Schmalensee, R. (2010): "Failure to Launch: Critical Mass in Platform Businesses", *Review of Network Economics*, 4(9), 1-26.

[10] Goettler, R. and Gordon, B. (2011): "Does AMD spur Intel to innovate more?", *Journal of Political Economy*, forthcoming.

[11] Gowrisankaran, G., Park, M. and Rysman, M. (2011): "Measuring Network Effects in a Dynamic Environment", working paper.

[12] Gowrisankaran, G. and Rysman M. (2011): "Dynamics of Consumer Demand for New Durable Goods", working paper.

[13] Hagiu, A. (2006): "Pricing and Commitment by Two-Sided Platforms", *RAND Journal of Economics*, 37(3), 720-737.

[14] Hendel, I. and Nevo, A. (2007): "Measuring the Implications of Sales and Consumer Inventory Behavior", *Econometrica*, 74(6), 1637-1673.

[15] Hu, Y. and Shum, M. (2011): "Nonparametric Identification of Dynamic Models with Unobserved State Variables," Working Paper.

[16] Imai, S., Jain, N., and Ching, A. (2009): "Bayesian Estimation of Dynamic Discrete Choice Models", *Econometrica,* 77(6), 1865-1899.

[17] Ishihara, M. and Ching, A. (2011): "Dynamic Demand for New and Used Durable Goods without Physical Depreciation: The Case of Japanese Video Games", Working Paper.

[18] Jiang, R., Manchanda, P., and Rossi, P. (2009): "Bayesian Analysis of Random Coefficient Logit Models Using Aggregate Data", J*ournal of Econometrics,* 149, 136-148.

[19] 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.

[20] Lee, Robin (2010): "Dynamic Demand Estimation in Platform and Two-Sided Markets", Working Paper

[21] McCulloch, R. and Rossi, P. (1994): "An Exact Likelihood Analysis of the Multinomial Probit Model", *Journal of Econometrics*, 64, 207-240.

[22] 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.

[23] Norets, A. (2009): "Inference in Dynamic Discrete Choice Models with Serially Correlated Unobserved State Variables", *Econometrica*, 77(5), 1665-1682.

[24] Pakes, A. and McGurie, P. (2001): "Stochastic Algorithms, Symmetric Markov Perfect Equilibrium, and the Curse of Dimensionality", *Econometrics*, 69(5), 1261-1281.

[25] Rochet, Jean-Charles, and Jean Tirole (2006): "Two-Sided Markets: A Progress Report." *The RAND Journal of Economics*, 35(3), 645-667.

[26] Rust, J. (1987): "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher." *Econometrics*, 55(5), 999-1033.

[27] Rysman, M. (2009): "The Economics of Two-Sided Markets." *Journal of Economic Perspectives*, 23(3), 125-143.

[28] Weyl, E. Glen (2010): " The Price Theory of Multi-Sided Platforms." *American Economic Review*, 100(4), 1642-1672.

A Substitution of the Software Market

In this paper, I assume that software products in the same submarket are substitutable and submarkets are separate from each other. Below I specify three different regression models to test the substitution between sports games, and the results are presented in Table 5.

Table 5: Empirical Results of Testing Software Competition Structure

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

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

In the first regression, the dependent variable is a game's price and the independent variables include: (i) the competition level within a submarket measured by the number of existing games in the same market; (ii) the competition from other submarkets measured by the number of existing games in all other submarkets; (iii) observed characteristics including the online rating score and the game age measured by the months after release; (iv) the market size measured by the log of the console owners; and (v) monthly dummies. The first important result is that the price of an existing game is lower by $0.148 if additional game is released in the same submarket and this impact is statistically significant. It implies that the competition within a market is strong. The second important result is that the competition effect from other submarkets is not statistically significant, which implies that submarkets are separate from each other. Besides, the game price is increasing in its online rating score, declining in its age and increasing in the number of console owners.

In the second regression, the dependent variable is the log of a game's 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 5 lists the estimation results which are consistent with the assumption of strong competition within a market and weak competition across markets.

The last regression mimics Nair (2007).[27] I still use the log of a game's unit sales as the dependent variable. However, I use the log of the total unit sales of all existing games in the same submarket to measure the competition level within a submarket, and the log of the total unit sales of all existing games in other submarkets to measure the competition effect from other submarkets. To address the endogeneity problem, I use the lagged price as an instrument for the current price, the number of existing games within the a submarket as an instrument for the within-submarket sales, and the number of existing games in other submarkets as an instrument for the outside-submarket sales. The results also show that the substitution effect within a submarket is strong while the substitution from games sold in other markets is insignificant.

[27]In 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.

B Computation

B.1 Jacobian Matrix

The Jacobian Matrix in equation (12) is

$$J_{(\mathbf{q}_{mt}\to\xi_{mt})} \equiv \parallel \nabla_{\xi_{mt}}\mathbf{q}_{mt} \parallel = \begin{bmatrix} \partial q_{1t}/\partial \xi_{1t} & ... & \partial q_{1t}/\partial \xi_{Jt} \\ \vdots & \ddots & \vdots \\ \partial q_{Jt}/\partial \xi_{1t} & \cdots & \partial q_{Jt}/\partial \xi_{Jt} \end{bmatrix}$$

with

$$\frac{\partial q_{jt}}{\partial \xi_{lt}} = \begin{cases} -\sum_{i=1}^{I} n_{mit}\left[b_{ijt}b_{ilt} + \beta_c b_{ijt} b_{im_0t}\frac{\partial EG_{mit+1}}{\partial \xi_{lt}}\right] & if\ l \neq j \\ \sum_{i=1}^{I} n_{mit}\left[b_{ijt}(1-b_{ijt}) + \beta_c b_{ijt} b_{im_0t}\frac{\partial EG_{mit+1}}{\partial \xi_{jt}}\right] & if\ l = j \end{cases}$$

Here, $b_{im_0t} = 1 - \sum\limits_{j\in J_{mt}} b_{ijt}$ is the probability of not purchasing. Notice that $\partial EG_{mit}/\partial \xi_{jt}$ and $\partial b_{im_0t}/\partial \xi_{jt}$ are determined by the following system of equations:

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

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 $\partial EG_{mit}/\partial \xi_{jt}$ for all i.

B.2 Best Response in Price

An incumbent software firm's problem is to pick a price to maximize the discounted profit:

$$\underset{\tilde{p}_{jt}}{max}\ \pi_j(\tilde{p}_{jt}, p_{-jt}, \mathbf{s}_{mt}) + \beta_f E\left[V_{jt+1}(\mathbf{s}_{mt+1}) \mid \mathbf{s}_{mt}, \tilde{p}_{jt}, \Psi_{-j}\right],$$

with

$$\begin{aligned}\pi_j(p_{jt}, p_{-jt}, \mathbf{s}_{mt}) &= (p_{jt} - c)\left[\sum_i n_{mit} b_{ij}(p_{jt}, p_{-jt}, \mathbf{s}_{mt})\right] \\ b_{ij}(p_{jt}, p_{-jt}, \mathbf{s}_{mt}) &= \frac{exp(x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt})}{exp[\beta_c EG_{it+1}(\mathbf{p}_{mt+1}, \mathbf{s}_{mt+1} \mid p_{jt}, p_{-jt}, \mathbf{s}_{mt})] + \sum_{j \in J_{mt}} exp(x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt})}.\end{aligned}$$

Below I show how to compute the marginal effect of current price on software firms' continuation values. Take an incumbent j's continuation value, $E\left[V_{jt+1}(\mathbf{s}_{mt+1}) \mid \mathbf{s}_{mt}, \tilde{p}_{jt}, \Psi_{-j}\right]$, for example. The state vector $\mathbf{s}_{mt}$ includes the number of existing games in the same submarket, the number of active high-type consumers, the number of active low-type consumers, the value of the No. 1 product in the same submarket, and its own consumption value. Notice that given competitors' prices and entrants' entry probabilities, a software's 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: $n_{mit+1} = n_{mit} b_{m0it} + Q_{mit}$, from which we can obtain the analytical form of $\partial \mathbf{s}_{mt+1}(p_{jt}, p_{-jt}, \mathbf{s}_{mt})/\partial p_{jt}$. Furthermore, along the estimation procedure, I approximated the value functions $V_{jt+1}(\mathbf{s}_{mt+1})$ by using polynomial regression in state variables. Therefore, I can pin down how current price affects the expectation of the next-period value function. A similar approach can be applied to computing the marginal effect of current price on consumers' continuation values.