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

Yiyi Zhou†

University of Virginia

This Version: November, 2011

(First Version: February, 2011)

Abstract

This paper studies the impact of pricing choices by platform intermediaries in a two-sided market with positive indirect network effects. It presents a dynamic equilibrium model to analyze consumers' purchase decisions for competing hardware platforms and affiliated software products, and software firms' dynamic pricing and entry decisions. This paper develops a new Bayesian approach for structural estimation of dynamic games. The estimation method is implemented on the U.S. fifth-generation video game industry (May 1995 - February 2002). The results show that overpricing one side of the market not only discourages demand on that side but also discourages participation on the other side, which over time can lead to a death spiral.

Keywords: Bayesian Estimation, Markov Chain Monte Carlo, Dynamic Game, Two-Sided Market, Indirect Network Effect, Forward-Looking Consumer, Dynamic Pricing and Entry

*I am extremely indebted to my advisers, Simon Anderson and Steven Stern, for their generous advice, guidance, and support. I am deeply grateful to my third reader, Federico Ciliberto, for his support, comments and suggestions. I also benefited from the conversations with Ambarish Chandra, Yongmin Chen, Andrew Ching, Hulya Eraslan, Andriy Norets, Regis Renault, John Rust, Holger Seig, Edward Vytlacil, Glen Weyl, Kenneth Wolpin, and Daniel Xu. I also thank Stephen Bruestle, Maxim Engers, Leora Friedberg, Nathan Larson, Toshihiko Mukoyama, Charles Murry, Christopher Otrok, John Pepper, Sarah Turner, Eric Young, and Nan Zhang, for their comments and suggestions. 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.

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

1 Introduction

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

This paper presents a dynamic equilibrium model to analyze consumers' purchase decisions for competing hardware platforms and their affiliated software products, and software firms' dynamic pricing and entry decisions. Consumers are heterogeneous, forward-looking, and have rational expectations about future software entry and prices. In each time period, they choose whether and when to purchase hardware and affiliated software. The hardware purchase decision and the software purchase decision are interdependent. On the one hand, the value of hardware depends on the value from being able to purchase the affiliated software. Hence, consumers rationally anticipate the software market when they make their purchase decisions of hardware. 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, incumbents decide how much to charge, and potential entrants decide whether to enter. There are four important dynamic factors that influence software firms' choices. First, the consumption value of a software product decays over time because consumers favor newness. Second, the competitive environment changes over time, which is mainly driven by the entry of new software products. Third, the distribution of potential buyers in each submarket changes over time. Because consumers are heterogeneous, they purchase hardware or enter the affiliated software submarkets at different points in time, and they also purchase software and leave a software submarket at different points in time. This gives software firms an incentive to engage in inter-temporal price discrimination. Fourth, consumers are forward-looking. If prices are declining and product variety is growing over time, consumers have an option value associated with waiting. Forward-looking software firms also take this into account when deciding their prices and entry.

In dynamic equilibrium models, given other agents' strategies, each agent's best response is the solution to a single-agent dynamic programming problem. Moreover, each equilibrium is the fixed point of the system of best response operators. For relatively complicated models, calculating the continuation value and policy function is computationally difficult, or even impossible. This paper provides a new practical method for structural estimation of these models. To the best of my knowledge, there is no published work on the use of Bayesian Markov Chain Monte Carlo (MCMC) methods for estimation of dynamic games. The difficulty with adopting a MCMC approach stems from the fact that estimating a dynamic game involves solving for the fixed point in the value function space as well as for the equilibrium in the agent action space.

The estimation method introduced by this paper combines the Bayesian algorithm and the dynamic game solution algorithm into a single algorithm that estimates the parameters and solves the model simultaneously. For a given draw of the parameter vector along the MCMC chain, I solve the dynamic game as follows: first, I randomly pick a subset from the entire state space for each period; second, for a given point in the subset, I nonparametrically approximate each agent's equilibrium strategy and value function by using the pseudo-best response functions and pseudo-value functions from previous MCMC iterations; third, 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 iterations. This algorithm 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.

My estimation approach has the following attractive features. First, it iterates each agent's Bellman equation only once for each draw of the parameter vector, and hence dramatically reduces the computational burden of estimating complicated dynamic games. Second, it fully solves the dynamic game, and hence can accommodate a rich specification of observed and unobserved heterogeneity. Third, the use of interpolation allows my estimation method to be able to handle models with large state spaces and continuous state variables.

The estimation method is implemented on the U.S. fifth-generation video game industry. This generation was dominated by three consoles, Sega Saturn, Sony PlayStation One and Nintendo 64. Sega Saturn failed during this period, even though it was very successful in the previous generation. My counterfactual simulations suggest that Sega priced inconsistently with the two-sided business pricing model and hence was shaken out of the market. It would have survived if it lowered its console price to attract more consumers and hence more games, or if it subsidized software R&D to encourage the participation of more games and hence the participation of more consumers, or if it lowered its royalty fees to reduce the game prices, attract more consumers and thus more games to join its platform.

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

Related Literature

This paper contributes the literature on two-sided markets which has been growing quickly in the last decade. Rysman (2009), and Hagiu and Wright (2011) provide general reviews of the literature on this field. One main result of previous theoretical studies is that pricing on one side of the market depends not only on its demand, but also on how it affects participation on the other side of the market (Rochet and Tirole, 2003; Anderson and Coate, 2005; Hagiu, 2006; Weyl, 2010). In addition, in those two-sided markets with positive indirect network effects, one side of the market is always waiting for the other before making its own action. This “chicken-and-egg” coordination problem is originally noted by Caillaud and Jullien (2003). However, most previous studies have usually adopted static models to investigate this problem which is really a dynamic game. In contrast, this paper presents a full dynamic game to look at the formation processes of the consumer network and the software network.

Not many empirical studies on two-sided markets exist. Rysman (2004) adopts a static model and estimates the network effects in the market for yellow pages. He finds that advertisers value consumer usage of a directory and consumers value advertising. Lee (2010) and Gowrisankaran, Park, and Rysman (2011) estimate forward-looking consumer demand for hardware and affiliated products. They mainly focus on the effect of the affiliated software products on the consumer hardware adoption, and do not model the decisions made by producers of affiliated products. In contrast, this paper models not only consumers' adoption decisions of hardware and software, but also software producers' price and entry decision. Therefore, this paper can examine not only how a platform's pricing on consumers affects the software market, but also how a platform's pricing on the software producers affects consumers' adoption of hardware.

This paper adds to the literature on dynamic pricing and the literature on dynamic entry. Most previous studies on dynamic pricing in the durable goods markets are theoretical studies which have focused on establishing closed-form results for relatively simple models. Few results exist for the more complicated multi-firm, multi-characteristic settings of actual markets. One exception is Nair (2007), who considers a video game seller as a single product durable-goods monopolist. In the literature on empirically estimating dynamic entry models, almost all previous studies have described the demand side in a static way (Ericson and Pakes, 1995; Bajari, Benkard and Levin, 2007). To my knowledge, no previous work has focused on both dynamic pricing and dynamic entry.

My estimation approach contributes to the literature on Bayesian estimation methods. This approach has been commonly applied to the static discrete choice models with latent variables.[1] Imai, Jain and Ching (2009), and Norets (2009) pioneered the use of Bayesian estimation method for dynamic discrete choice models. Both use the MCMC algorithm to draw a sequence of parameter vectors from their posterior distributions. During each MCMC iteration, they partially solve for the value functions and stores those partially solved value functions. For the current trial parameter vector, they non-parametrically approximate the expected value functions using those solved value functions from past MCMC iterations. They also provide theory to justify statistical inference made based on this algorithm. My method is based on their idea: iterate the Bellman equation only once at each estimation iteration and use the outcomes from previous iterations to approximate the expectation in the Bellman equation. In contrast to those two papers, my estimation method is designed for estimating dynamic games which have another layer of complication because each equilibrium is the fixed point of the best response system.

My estimation approach contributes to the literature on estimating dynamic games. The most popular methods in the literature are the nested fixed point approach (for example, Berry, Levinsohn, and Pakes, 1995) and the two-step approach (for example, Bajari, Benkard, and Levin, 2007). The former solves for the equilibrium for each guess of parameter vector. The latter sidesteps the equilibrium computation step by substituting nonparametric functions of the data for the policy

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

functions. In contrast to the existing approaches, the Bayesian method proposed in this paper uses MCMC algorithm to simulate the posterior distribution of the structural parameters, and solves the dynamic game for each draw of parameter vector by incorporating nonparametric approximation method and interpolation method.

2 The U.S. Videogame Industry

Since Pong was introduced in the early 1970's, the U.S. video game industry has grown to reach 22 billion dollars in revenue in 2008, over twice the total box-office revenue in the movie industry (10 billion dollars). The video game industry is a two-sided market in which consoles (hardware) act as platform intermediaries, and consumers and video games (software) are on the two sides of the market. On one side, console providers design and sell consoles to consumers who pay a one-time fixed fee to join a platform (i.e., the console price). On the other side, console providers charge game producers a royalty fee for the right to the code which allows the 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 per copy sold to consumers. In fact, console providers manufacture all the video games themselves to track sales for royalty collection.

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

2.1 Data

My main data set is obtained from the NPD Group, a market research firm. It includes the monthly revenue and unit sales of three fifth-generation consoles, Sega Saturn, Sony PlayStation One (PS1) and Nintendo 64 (N64), from May 1995 until February 2002. I calculate the console price by taking the ratio of revenue over unit sales in each month. 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 the unit sales for 1697 unique game titles released for the three consoles during this period. It was collected from thirty of the largest retailers in the U.S., which account for around 85% of video game sales, and was extrapolated by the NPD for the entire U.S. market. I calculate the game price by taking the ratio of revenue over unit sales in every month. The data that I use for estimating the game market only includes sports games. I do this because it is relatively easy to sort sports games into groups and it reduces the estimation time to use a smaller sample. The data for estimation contains 397 sports games distributed in 29 software submarkets. I also collect the data of the critics and user rating score 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 U.S. market for the Saturn, PS1, 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 Sega Saturn was the first mover, it 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 three 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.

1. Positive Indirect Network Effect

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. Hence, the number of users of a console is largely contingent on current and expected availability and prices of games; and the number of games affiliated with a console depends on how many users have purchased and are expected to purchase that console.

On the one side of the market, consumers decide whether to purchase consoles and games. A console has no stand-alone value. Its value comes from the compatible game titles. Figure 1 (a) presents the number of each console's owners during the sample period. The installed bases of PS1 users and N64 users grew fast during this period. On the contrary, the number of Saturn owners stopped growing one and a half years after its release, because consumers stopped buying it.

On the other side of the market, incumbent game producers choose their prices and potential entrants choose whether to enter. Figure 1 (b) presents the number of existing game titles sold for each console in every month during the sample period. The number of PS1 game titles and the number of N64 game titles grew fast. By contrast, the number of Saturn game titles started to shrink from January 1998, because no new games affiliated with that platform.

2. Game Prices and Sales Decline with Age

The second feature of the U.S. video game market is that game price and sales start at a high level, then decline rapidly in the first six months after release. In figure 2, 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 is around \$45 per copy at release and then drops to about \$23 in the preceding year. The average game unit sales are around 40 thousand in the first month and then fall to around 5 thousand per month after the first year.

What drives the game price 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 marker and production cost. Both costs remain roughly constant per unit over time.[2] The most reasonable explanation is inter-temporal price discrimination. Consumers are heterogeneous in their preferences for either product characteristics or price or both. They purchase consoles and games at different times. 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 cut prices thereafter to appeal to those with low net valuations. In addition, the entry of new games leads to more intense competition and thus induces the existing game titles to cut their prices.

3. Seasonality

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

3 Model Framework

In this section, I present a dynamic equilibrium model of demand and supply. The model is dynamic, time is discrete and the horizon is finite.[3] There exists a finite number of hardware platforms.

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

[3] In the application to the video game industry, I focus on one generation market. My data set is monthly data lasting for 82 months, hence I set one time period is one month. I assume that one generation dies after 100 time

Consumers with no hardware decide whether to buy one in each time period. Each consumer is allowed to buy at most one hardware.[4] Once she owns one, she does not need to make the purchase decision of hardware any more. Moreover, once she own a hardware, she become a potential buyer for the affiliated software products.

The software market consists of M separate submarkets, explicitly ruling out competition across submarkets. Each consumer can only purchase at most one software product within a submarket, explicitly allowing for competition within a submarket. In the context of videogames, I define a software submarket that a game title belongs to based on the console that game is compatible with and the game genre it is grouped in.[5] In Appendix B, I find supporting evidence for this assumption as sports video games are found to be strong substitutes within a submarket and weak substitutes across submarkets.

The following events occur in each software submarket and in each time period:

(i) Each incumbent software producer decides how much to charge. Each potential entrant draws an entry cost from a known distribution, and decides whether to enter. 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 software. Each consumer is allowed to buy at most one product in a software submarket. Hence, 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 producers' dynamic pricing and entry, and lastly define 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. The expected

periods (roughly 8 years).

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

[5]For example, Football games on PlayStation 1 is a submarket, Baseball games on PlayStation 1 is a submarket, Football games on Nintendo 64 is another submarket, and so on.

lifetime utility of consumer type i from purchasing platform l at time t is

$$U_{ilt} = \Gamma_{ilt} - \varphi_i^H P_{lt} + \psi^H X_t + \xi_{lt}^H + \epsilon_{ilt}^H,$$

where Γ_{ilt} is 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. P_{lt} is the price of hardware l, and φ_i^H represents consumer type-specific sensitivity to money. In this paper, I focus on how hardware providers' choices affect consumers' purchase decisions and software producers' pricing and entry decisions. I do not model how hardware providers make those choices; rather, I treat hardware price as exogenous.[6] X_t is the holiday dummy. ξ_{lt}^H represents additional hardware characteristics observed by consumers but not by researchers. ϵ_{ilt}^H 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:

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

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}^H and ϵ_{i0t}^H are assumed to follow the standard Type-I Extreme Value distribution and are $i.i.d.$ over time, products and consumer types.

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

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

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

[6]It is widely speculated that all the major consoles were sold at a price near marginal cost. The literature on two-sided markets (e.g., Armstrong 2006 and Hagiu 2006) provides good reasons why a platform provider keeps the price of one side low and make money from the other side. Hence, the declining console price may be mainly driven by the declining costs of producing consoles.

demand shocks ϵ_{it}^H:

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

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

$$H_{it}(\mathbf{S}_t) = ln\{\sum_l exp(\Gamma_{ilt} - \varphi_i^H P_{lt} + \psi^H X_t + \xi_{lt}^H + \xi_{lt}^H) + exp[\beta_c EH_{it+1}(\mathbf{S}_{t+1} \mid \mathbf{S}_t)]\}. \tag{1}$$

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

$$B_{ilt}(\mathbf{S}_t) = \frac{exp(\Gamma_{ilt} - \varphi_i^H P_{lt} + \psi^H X_t + \xi_{lt}^H + \xi_{lt}^H)}{exp[\beta_c EH_{it+1}(\mathbf{S}_{t+1} \mid \mathbf{S}_t)] + \sum_L exp(\Gamma_{ilt} - \varphi_i^H P_{lt} + \psi^H X_t + \xi_{lt}^H + \xi_{lt}^H)}. \tag{2}$$

Hence, the demand for the hardware l in period t is $Q_{lt} = \sum_i N_{it} B_{ilt}$, where N_{it} is the number of consumers who have not purchased any hardware at time t. In 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

As I assumed, the software market consists of M separate submarkets. Below I describe consumers' demand for software, and software producers' pricing and entry in a software submarket. Same events occur in other submarkets.

Software Utility

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

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

where x_{jt} is a vector of observed software product characteristics, including platform-specific dummy, online rating score, product newness and holiday dummy; 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 consumers preferences in observed software characteristics and

φ_i is type-i consumer's sensitivity to money.

The utility of not buying in the submarket m at time t as the sum of the discounted expected value of waiting and an idiosyncratic consumer taste:

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

where ϵ_{im_0t} is the idiosyncratic taste from not buying. ϵ_{ijt} and ϵ_{im_0t} are assumed to follow the standard Type-I Extreme Value distribution and *i.i.d.* over time, products and consumer types.

Consumer Belief

Most of the literature on estimating dynamic demand models assumes that consumer purchase decisions are only based on a scalar state variable (the inclusive value) which follows an AR(1) process.[7] Such a functional form restriction on 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 in theory literature and can be reconciled with a consistent supply model.

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. Besides, consumers also can observe the price of each available product, $\mathbf{p}_{mt} \equiv \{p_{jt}\}_{j \in J_{mt}}$, and their own demand shocks in submarket m, $\epsilon_{mit} = (\{\epsilon_{ijt}\}_{j \in J_{mt}}, \epsilon_{im_0t})$.

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

Software Purchase

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

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

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

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

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

Consumer Distribution

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

$$n_{mit+1} = n_{mit}(1 - \sum_{j \in J_{mt}} b_{ijt}) + Q_{mit} \tag{5}$$

where $n_{mit}(1 - \sum_{j \in J_{mt}} b_{ijt})$ is the mass of consumers who do not buy in period t and remain active the next period; and Q_{mit} is the mass of new consumers who purchase the compatible hardware, as described in the previous subsection.

Total Software Utility

In the previous subsection, I specify that the consumption value of a hardware depends on the total utility from being able to purchase its affiliated software, Γ_{ilt}. To close the demand side of the model, I need to link it to the value of being able to purchase the affiliated software. Let M_l denote the set of software submarkets affiliated with hardware l. Then, Γ_{ilt} is a type-i consumer's total value being active in all submarkets affiliated with hardware l:

$$\Gamma_{ilt} = \sum_{m \in M_l} G_{it}(\mathbf{s}_{mt}, \mathbf{p}_{mt}). \tag{6}$$

3.3 Software Pricing and Entry

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

3.3.1 A Firm's Problem

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

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

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

Potential Entrants. Every period, there is finite number of potential entrants outside the software submarket m. Let E_{mt} denote the set of potential entrants. Each potential entrant $j \in E_{mt}$ first draws an entry cost from a known distribution and then decides whether to enter. Potential entrants are short lived and base their entry decisions on the net present value of entering today; they do not take the option value of delaying entry into account. If it enters, it pays the entry cost and starts to earn profit next period; if not, it earns zero profits. The entry cost is assumed to be $\gamma_m + \nu_{jt}$ where γ_m is the component that is common to all software firms in submarket m and ν_{jt} is a private information shock which is assumed to be independently and identically distributed across firms and periods with c.d.f. $F_\nu(\cdot)$. Let $y_{jt+1} = 1$ iff entrant j enters. Essentially, a potential entrant j's optimization problem is to compare the entry cost and the expected profit.

3.3.2 Perceived Strategy Function

Because 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 firms. The game has a Markov structure, and I assume that each firm plays Markov strategies. That is, if $\mathbf{s}_{mt} = \mathbf{s}_{m't}$, then firm j's decision in submarket m and m' are the same. Let $\Psi = \{\Psi_{jt}(\mathbf{s}_{mt})\}$ 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.

3.3.3 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

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

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

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

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

3.3.4 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:

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

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

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

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

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

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

$$
\begin{aligned}
V_{jt}^{e}(\mathbf{s}_{mt} \mid \Psi) &= \max_{\tilde{\rho}_{jt}} - \int_{\nu_{jt} < F_{\nu}^{-1}(\tilde{\rho}_{jt})} \nu_{jt} dF_{\nu}(\nu_{jt}) \\
&\quad + \tilde{\rho}_{jt} \left\{ -\gamma_m + \beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \Psi) \mid \mathbf{s}_{mt}, \Psi] \right\}. \qquad (8)
\end{aligned}
$$

3.4 Equilibrium Concept

In this model, a hardware's value depends on the total utility from affiliated software, and thus any software firm's choice affects consumers' adoption of hardware. However, to simplify the model, I assume that software firms do not take that effect into account when they make their choices. One condition is that each software is tiny compared to the whole market. Under this assumption, strategic interactions only occur among software firms in the same submarket.

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) - (8) are simultaneously satisfied at every state $\mathbf{s}_{mt}$. In other words, the equilibrium is the fixed point of the game defined by equation (1) - (8), with the following properties.

[1] Software Firms. Equation (7) implies that in equilibrium, when faced with state $\mathbf{s}_{mt}$, each incumbent software producer's pricing policy is a best response to other software firms' strategies and consumers' behavior at that state. Meanwhile, equation (8) 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.

[2] 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 intertemporal utility. In addition, the value of a hardware is given by the equation (6).

[3] State Transition. Software firms take into account the effect of their actions on the evolution of states in the submarket through equation (5).

4 Bayesian Estimation

In this section, I describe my estimation method in detail. Let θ denote the vector of parameters in the model that need to be estimated. Let *data* denote all the data available for estimation which includes two parts: (i) the 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

Generally speaking, it is difficult to analytically prove the existence and uniqueness of a MPE in pure strategy for dynamic oligopoly models.[8] I am unable to formally state whether an equilibrium exists and whether the equilibrium is unique.[9] In this paper, I assume that there exist a unique equilibrium.

Let $\mathcal{L}(data \mid \theta)$ denote the likelihood. Rather than using a nested-fixed point or a two-step method to maximize the likelihood, I employ the Bayesian MCMC method to sample the parameter vector θ from its posterior distribution,

$$\mathcal{P}(\theta \mid data) \propto \mathcal{L}(data \mid \theta)\pi(\theta), \tag{9}$$

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

4.2 Likelihood Contributions $\mathcal{L}(data \mid \theta)$

The demand for hardware is a dynamic discrete choice model. I assume that the unobserved (to researcher) platform-specific demand shifters ξ_{lt}^{H} are normally distributed with mean zero and variance $\sigma_{\xi H}^2$, independent across all products and over time. The distribution of the aggregate

[8] One way is to impose additional restrictions on the model to ensure that each agent's best reply is always unique. Doraszelski and Satterthwaite (2010) provide a condition on model primitives that guarantees the existence of pure strategy equilibrium in Ericson-Pakes-style models. To guarantee the uniqueness of the entry and exit best responses, they assume that the densities of the scrap values and setup values are continuous. Furthermore, they prove that if the transition function is unique investment choice (UIC) admissible, then a firm's investment decision is indeed uniquely determined.

[9] I have proved that, under some restrictions, there exists a unique equilibrium in pure strategy for a finite-period model of dynamic oligopoly pricing with forward-looking consumers. Yet it is extremely hard to go further to show the equilibrium existence for the model proposed in this paper which also contains dynamic entry and self-selected consumers.

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

$$\mathcal{L}_d^H(\mathbf{Q}_t \mid \mathbf{S}_t;\theta) = \prod_l \left[\phi(\xi_{lt}^H/\sigma_{\xi H})/\sigma_{\xi h}\right] \mid \left(J_{(\mathbf{Q}_t \to \xi_{lt}^H)}\right)^{-1} \mid, \tag{10}$$

where $\phi(\cdot)$ is the *pdf* of the standard normal distribution and $J_{(\mathbf{Q}_t \to \xi_{lt}^H)}$ 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.[10] 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

$$\mathcal{L}_d^G(\mathbf{q}_{mt} \mid \mathbf{s}_{mt}, \mathbf{p}_{mt};\theta) = \prod_{j \in J_{mt}} \left[\phi(\xi_{jt}/\sigma_\xi)/\sigma_\xi\right] \mid \left(J_{(\mathbf{q}_{mt} \to \xi_{mt})}\right)^{-1} \mid, \tag{11}$$

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

Next I specify the likelihood contribution of the software pricing policy function. Let $\tilde{p}_{jt}$ and p_{jt}^* denote the observed price and the actual price of product j at time t, respectively. Assume that the observed price is proportional to the actual price, that is, $\tilde{p}_{jt} = p_{jt}^* \varsigma_{jt}$, where ς_{jt} is the measurement error that reflects discrepancies between the observed prices and the actual prices.[11] Furthermore, it is assumed to follow a log-normal distribution with mean zero and variance σ_ς^2, 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

$$\mathcal{L}_p(p_{jt} \mid \mathbf{s}_{mt};\theta) = \frac{1}{\sigma_\varsigma}\phi\left(\frac{ln[\tilde{p}_{jt}/p_j(\mathbf{s}_{mt},\theta)]}{\sigma_\varsigma}\right). \tag{12}$$

To specify the likelihood contribution of the software entry policy function, I assume that the entry cost shocks follow an independent normal distribution with mean zero and variance σ_ν^2.[12] Hence,

[10] 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.

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

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

conditional on the state vector $\mathbf{s}_{mt}$, the likelihood contribution of entrant $j \in E_{mt}$ is

$$\begin{aligned}\mathcal{L}_y(y_{jt+1} \mid \mathbf{s}_{mt};\theta) &= \left(\Phi\left[\frac{\beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \mathbf{s}_{mt};\theta)] - \gamma}{\sigma_\nu}\right]\right)^{y_{jt+1}} \\ &\quad \times \left(1 - \Phi\left[\frac{\beta_f E[V_{jt+1}(\mathbf{s}_{mt+1} \mid \mathbf{s}_{mt};\theta)] - \gamma}{\sigma_\nu}\right]\right)^{1-y_{jt+1}}. \end{aligned} \tag{13}$$

Therefore, the likelihood can be written as

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

4.3 Estimation Algorithm

My estimation algorithm involves two loops (see Appendix A): in the outer-loop, I use Metropolis-Hastings algorithm to update the structural parameters; and in the inner-loop, for a given parameter vector, I solve the dynamic game by incorporating the non-parametric approximation method. Below I describe the two loops in detail.

4.3.1 Outer-Loop: Metropolis-Hastings

The posterior distribution in equation (9) is a high-dimensional and complex function of the parameters. It is known that, instead of drawing the entire parameter vector at once, it is often simpler to partition it into blocks and draw the parameters of each block separately given the other parameters. Based on the model, I partition all parameters into 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, $\theta_1 = (\psi^H, \varphi_i^H, \sigma_{\xi H})$; (ii) the first 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 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, $\theta_4 = (\gamma_{Saturn}, \gamma_{PS}, \gamma_{N64}, \sigma_\nu)$.

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 proposal density. As usual in the literature,[13] 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 (10), (11), (12) and (13), 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, we accept the candidate parameter vector $\theta_l^{*(k)}$ with probability $min\{\eta_l^{*(k)}, 1\}$.

4.3.2 Inner-Loop: Dynamic Game Solution

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 introduce a new method of solving the dynamic game suitable for use in conjunction with the Bayesian MCMC estimation.

For a given draw of parameter vector, the main procedure goes as follows. First, for a given point of the state space, I non-parametrically approximate the equilibrium strategies and value function using the pseudo-best responses and pseudo-value functions of past iterations. To obtain each agent's continuation value that requires to solving for the value function at all possible states, I adopt interpolation method to deal with the large state space problem. Then, I solve for each agent's best response and value functions given other agents play the approximated equilibrium strategies, and store those pseudo-best responses and pseudo-value functions for future iterations.

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

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.[14] 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.[15]

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 firm j in iteration k is computed as

$$\hat{\Psi}_{jt}^{(k)}(\mathbf{s}_{mt}, \theta) \quad = \quad \sum_{n=1}^{N(k)} \Psi_{jt}^{(k-n)}(\mathbf{s}_{mt}, \theta^{*(k-n)}) \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})}, \tag{14}$$

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 firm's optimization problem, $p_{jt}^{(k)}(\mathbf{s}_{mt}, \theta)$, and Appendix D presents the computation method in detail. For entrant j, the pseudo-best response in entering probability is the solution to the potential entrant's optimization problem. Under the assumption of normally distributed entry cost shocks, it is

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

In essence, the equilibrium strategies is 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.[16]

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

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

[16]Here, I need to develop formal proofs. It is part of future research.

Non-Parametric Approximation of Value Function

To compute the expected value function at a given state point, the conventional iterates 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. 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 expectation of the next-period value function at next-period state $(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1})$ in iteration k is approximated as

$$\hat{E}\left[G_{it+1}^{(k)}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1}, \theta)\right] = \sum_{n=1}^{N(k)} G_{it+1}^{(k-n)}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1}, \theta^{*(k-n)}) \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})}, \qquad (15)$$

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 $\hat{\Psi}^{(k)}$

$$\begin{aligned} G_{it}^{(k)}(\mathbf{s}_{mt}, \mathbf{p}_{mt}, \theta) \quad &= \quad ln\{\sum_{j \in J_{mt}} exp(x_{jt}\psi - \varphi_i p_{jt} + \xi_{jt}) \\ &\quad +exp\left(\beta_c \hat{E}[G_{it+1}^{(k)}(\mathbf{s}_{mt+1}, \mathbf{p}_{mt+1}, \theta) \mid \mathbf{s}_{mt}, \mathbf{p}_{mt}, \hat{\Psi}^{(k)}, \theta]\right)\} \end{aligned} \qquad (16)$$

The approximated expected value function given by equation (15) is the weighted average of the pseudo-value functions of $N(k)$ most recent iterations. IJC (2009) show that, as the MCMC iterations and the number of past iterations for approximating the expected value functions increase, the pseudo-value function converges to the true value functions, and the posterior parameter draws based on the pseudo-value functions will also converge to the true posterior distributions. Moreover, the convergence of the approximated expected value function to the true value function requires that $N(k) \to \infty$ and $k - N(k) \to \infty$ as $k \to \infty$.

A similar method applies to computing the expectation of a software producer's next-period value function

$$\hat{E}\left[V_{jt+1}^{(k)}(\mathbf{s}_{mt+1}, \theta)\right] = \sum_{n=1}^{N(k)} V_{jt+1}^{(k-n)}(\mathbf{s}_{mt+1}, \theta^{*(k-n)}) \times \frac{K_h(\theta - \theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta - \theta^{*(k-n)})} \tag{17}$$

where $V_{jt+1}^{(k)}(\mathbf{s}_{mt}, \theta)$ is the incumbent software firm j' pseudo-value function at state $\mathbf{s}_{mt}$ conditional on that all other software firms playing the equilibrium $\hat{\Psi}^{(k)}$:

$$V_{jt}^{(k)}(\mathbf{s}_{mt}, \theta) = \max_{\tilde{p}_{jt}} \pi_{jt}(\mathbf{s}_{mt}, \tilde{p}_{jt}, p_{-jt}) + \beta_f \hat{E}\left[V_{jt+1}^{(k)}(\mathbf{s}_{mt+1}, \theta) \mid \mathbf{s}_{mt}, \tilde{p}_{rjt}, \hat{\Psi}^{(k)}, \theta\right]. \tag{18}$$

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 (16) and equation (18), we still need to compute equation (15) and (17) for every possible point of the state space. Due to the "curse of dimensionality",[17] 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 approach to deal with the large state space problem. In the first step, I randomly choose a subset of the state points every period, and obtain the expected values at those points with the non-parametric approximation approach described above. Next, I interpolate the value functions with a quadratic-in-states polynomial

[17] 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 $(KM)^N$.

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 finite times and then take the average of the predicted values. This estimation method 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 method significantly alleviates the computational burden and makes it possible to estimate models with very large state spaces and rich structure. However, we also should notice that estimators of structural parameters are not consistent as long as interpolation is used, because the approximation errors in the expected value functions enter nonlinearly in optimization problems.[18]

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. Besides, 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 choice 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 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 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 incumbent firms do. The only difference is that they trace the expected value of new product instead of the value of its own product.

[18]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.

Computing ξ_{lt}^H 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 (10), I update the aggregate demand shocks based on the expression,

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

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

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

5 Estimation Results

5.1 Econometric Details

Consumer Heterogeneity. For simplicity but without loss of generality, I assume two consumer types, high-type consumers and low-type consumers, with different sensitivity to price.[19] Besides, it is necessary to choose an initial number of consumers, N_0. Once this is pinned down, the future distribution of each consumer type is determined by the consumer purchase decisions of hardware and software. In particular, their 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 choose N_0 to be 100 millions.

[19] 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.

Prior Setting. In order to estimate the model it is necessary to specify the prior distribution for the parameters and the equilibrium probabilities to be estimated. Consumer's preference to product characteristics (ψ and ψ^H) and consumer's sensitivity to price (φ_i and φ_i^H) follow normal distribution with mean zero and large standard deviation. 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 are log-normal distribution with mean zero and large standard deviation.

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

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

5.2 Estimates

I draw 100,000 samples from the posterior distribution. But, I use the last 50,000 samples to derive the posterior means and standard deviations. They are reported in table 3. In addition, I also compute those statistics from the last 25,000 samples. Since they are not statistically different, I conclude that the samples that I use to compute the posterior statistics are drawn from a stable distribution.[20]

The estimates in the consumer utility function of hardware are consistent with our expectation. High-type consumers' price sensitivity is 0.0018 and low-type consumers' price sensitivity is 0.0064. They are positive just 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

[20] Here, I plan to do more stability tests.

good gifts during holiday season. Besides, High-type consumers correspond to 14.6% 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 it is more technologically advanced than the other two. Consumers favor the games with high online rating score. They dislike games which have been in the market for a long time, partly because most sports games are designed based on the latest tournaments. 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 consumers' price sensitivity is 0.0137 and low-type consumers' price sensitivity is 0.0507.

The cost per unit is \$14.7, \$10.8 and \$18.5 for games released on Sega Saturn, PS1 and N64, respectively. As Coughlan (2001) reported, the production/packaging cost for 32-bit CD-ROM games is around \$1.5 per disc. Hence, the royalty fee charged by Sega and Sony were \$13.2 and \$9.3 per copy sold. The unit cost of N64 games is much higher than Saturn games and PlayStation games, because Nintendo used ROM cartridges to store games whose production expense was much higher than compact disc format used by competitors.

The average entry cost of Saturn games, PS1 games and N64 games are 4.7 million dollars, 3.7 million dollars and 4.6 million dollars, respectively. The standard deviation of entry cost is 2.5 million dollars. The R&D cost of Saturn games was on average significantly higher than PS1 games, even though both adopted very similar technology, partly due to Saturn's dual-CPU architecture and its more complex graphics.[21]

5.3 Fit of Model

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

Figure 5 (a) compares the predicted and the observed price across all sports games at each age (i.e., months since introduction). It indicates that the proposed model is able to explain the

[21]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.

declining pattern of game price. However, the predicted price drops not so quickly as the observed price. One possible reason is that the model does not consider the second-hand market which contributes to the declining game price in the data. 5 (b) compares the predicted and the observed unit sales across all sports games at each age. It shows that the model fits the data very well.

5.4 Software Pricing

This subsection investigates the importance of each factor that influences competing game producers' price choices. To achieve this goal, I approximate the pricing policy function as a polynomial of observed and unobserved state variables of the equilibrium model.

In the polynomial regression, the dependent variable is software price. The state variables include: (i) number of active software products in the same submarket which measures the competition level, (ii) percentage of high-type consumers remaining in a submarket which measures the consumer composition, (iii) consumer waiting values which measures consumer forward-looking behavior, and the consumption value which measures a product's quality.[22]

Table 2 reports the first-order and second-order polynomial regression estimates, which implies four important results. First, if a new software product is introduced to the same submarket, the price of the existing software products are lower. Second, software price is increasing in the proportion of high-type consumers. Third, software price is decreasing in the high-type consumers' waiting value, and increasing in the low-type consumers' waiting value. Fourth, software price is increasing in its own consumption value.

6 Counterfactuals

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 is the impact of platform pricing on the number of consumers, the variety and prices of the affiliated products?

In the video game industry, console platforms choose console prices, software royalty fees, and software research and development fees. On the consumer side of the market, the console price is a one-time payment for access to the platforms. On the software side of the market, royalty fee is a fixed payment per unit sold to consumers. In addition, game producers also have to spend a one-time payment of research and development if they want to join a platform. Even though

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

this cost does not directly go to platform providers' pockets, they can still strategically influence this software entry fees either by offering easier-learning development technology or by subsidizing software R&D directly.

6.1 Sega Saturn's Failure

Sega went from accounting for around 40% of the worldwide console sales in the fourth-generation of video games (November 1998 - May 1995) to around 6.5% of the worldwide console sales in the fifth-generation of video games (May 1995 - November 2000).[23]

What happened to Sega Saturn? It was released at a high price of $399, $100 higher than PlayStation ($299) and $200 higher than Nintendo 64 ($199). In two-sided markets, a high price on consumer side discourages the entry of consumers, and hence the entry of software producers. When there are multiple competing platforms, the effect of participation of one side on the other has even more bite. High console prices lose consumers to the competing platforms, which upgrades the value of the competitors to consumers, and hence leads to a large decrease in buyer interest in the original platform. In addition, it was widely agreed that Saturn was difficult to develop compared to PlayStation.

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

6.2 Console Price

Figure 4 (a) presents the prices of the three fifth-generation consoles in every month during the sample period. Console prices were generally declining over time. It also shows that Sega's console prices were $100 higher than its competitors at the same age for the first two year. A high console price discouraged consumer entry and hence software entry. It is speculated that contributes to the

[23] In the fourth-generation video game market, two major console platforms dominated the fourth-generation. The biggest one was Super Nintendo Entertainment System, launched in November 1990 in Japan. It sold 49.1 million units worldwide. The other big platform was Sega Mega, launched in November 1998, which sold 40 million units worldwide. The total unit sales of other small platforms was around 10 million worldwide. In the fifth-generation market, the worldwide unit sales for the big-3 console platforms, Sega Saturn, Sony PlaySation and Nintendo 64, are 9.5 million, 102.5 million and 32.9 million, respectively. Sega lost behind far away from its rivals.

failure of Sega.[24] In this counterfactual, I consider what if Sega has followed the price schedule of its main rival, Sony PlayStation One. The goal is to investigate whether Sega could survive if it has charged this lower console price schedule.

With the proposed price schedule for Sega, Sega would increase its console owners by around 10M. A lower console price attracts more users and hence more games, which attract more users and so on. Meanwhile, PlayStation One would suffer from Sega's lower console price strategy, but Nintendo 64 would not affected much. PS1's technology and entry time were almost the same as Saturn, and hence was the main rival of Saturn; however, N64 was more technologically advanced and entered the market two years later than Saturn, and hence was not Saturn's direct competitor.

6.3 Subsidize Software Entry

Compared to PlayStation, Saturn is difficult to develop for. High R&D cost discourages software entry and thus discourages consumer entry. In this counterfactual, I consider what if Sega had given game producers a subsidy of $1M per game.

(incomplete)

6.4 Royalty Fees

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

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

(incomplete)

7 Conclusion

In this paper, I proposed a new Bayesian MCMC approach for structural estimation of dynamic games. I used the outcomes from past MCMC iteration to approximate each agent's equilibrium

[24] See wikipedia, http://en.wikipedia.org/wiki/Sega_Saturn#cite_note-16.

strategy and value function for the current draw of parameter vector. This significantly reduces the computation burden. To avoid computing the value function at all possible points of the state space, I provided an approach which combines the nonparametric approximation method and interpolation method. I demonstrated the effectiveness of my method by applying it to a dynamic model of pricing and entry in the video game market. The method here can be applied to estimate other dynamic games, especially those with unobserved heterogeneity and large state space.

This paper developed a framework to study pricing and entry in a two-sided market with positive indirect network effects. On the demand 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 not only accounts for dynamic selection of forward-looking and heterogeneous consumers into platforms for affiliated software products, but also allows for the contingency of platforms' value on the availability and prices of the affiliated products. Incorporating the complementarity between hardware and software, I am able to examine how a change in the pricing on the software side affects consumers' entry to a platform. My counterfactual experiments showed that if Sega had subsidized software entry it would have substantially encouraged more software entry and hence more users. In addition, my counterfactual experiments showed that if Sega had lowered game entry cost it would have boosted game entry and hence consumer entry.

On the supply side, software firms compete within a submarket. Incumbents choose their prices and potential entrants choose whether to enter, strategically accounting for competitors' reactions and consumers' responses. I have investigated how a change in the pricing on the consumer side affects software firms' entry and price choices. Counterfactual experiments demonstrated that lowering console price has significantly increased consumer entry and hence software entry.

References

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

[2] Anderson, S. and Coate, S. (2005): "Market Provision of Broadcasting: A Welfare Analysis", *Review of Economic Studies*, 72(4), 947-972.

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

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

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

[12] Hagiu, A. and Wright, J. (2011): "Multi-Sided Platforms", Working Paper.

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

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

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

[16] 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.

[17] 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.

[18] 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.

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

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

[21] 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.

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

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

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

[25] Rysman, M. (2004): "Competition Between Networks: A Study of the Market for Yellow Pages." *Review of Economic Studies*, 71, 483-512.

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

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

Tables:

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

	Sega Saturn	PlayStation	Nintendo 64
HARDWARE			
Release Date	May 1995	Sept. 1995	Sept. 1996
Provider	Sega	Sony	Nintendo
CPU bits	32	32	64
MHZ	28	33.87	93.75
Starting price	\$399.9	\$299.7	\$199.8
Ending price	\$41.0	\$112.2	\$87.1
Average unit sales per month (million)	0.02	0.36	0.26
Installed base (Feb. 2002, million)	1.28	28.25	17.17
SOFTWARE			
Total active titles	240	1172	385
Total unit sold (Feb. 2002, 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)

Note: Numbers in parenthesis are standard deviations. Data source: NPD group.

Table 2: Polynomial Regression of Pricing Policy Function

State Variables	First-Order	Second-Order
s_1 (# competitors)	-0.3241***	-0.0336
s_1 square		0.0036***
s_2 (% high-type consumers)	0.2734 ***	1.2489***
s_2 square		-0.0202***
s_3 (high-type waiting value)	-1.4450 ***	-0.8588***
s_3 square		0.0668***
s_4 (low-type waiting value)	1.3430***	0.7073***
s_4 square		-0.0788***
s_5 (its consumption value)	1.5168 ***	2.2909***
s_5 square		-0.0729***
R-square	0.47	0.57

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

Table 3: Posterior Means and Standard Deviations

	Last 50,000 Samples		Last 25,000 Samples	
	Mean	Std dev	Mean	Std dev
Block 1: Demand for Hardware				
φ_1^H (H-type consumer price sensitivity)	0.0018	0.0000	0.0018	0.0000
φ_2^H (L-type consumer price sensitivity)	0.0064	0.0002	0.0064	0.0002
ψ_{Nov}^H (Nov. dummy)	0.7454	0.0217	0.7464	0.0219
ψ_{Dec}^H (Dec. dummy)	2.4049	0.0904	2.3991	0.0903
$\sigma_{\xi H}$ (std of hardware demand shocks)	0.1049	0.0058	0.1058	0.0061
δ (initial share of H-type consumers)	0.1460	0.0009	0.1460	0.0009
Block 2: Demand for Software				
ψ_{N64} (dummy for N64 games)	1.5444	0.0343	1.5372	0.0074
ψ_1 (online rating score of games)	0.1689	0.0090	0.1712	0.0068
ψ_2 (game age if new)	-0.3333	0.0097	-0.3407	0.0023
ψ_3 (game age if old)	-0.1894	0.0062	-0.1931	0.0033
ψ_{Nov} (Nov. dummy)	0.2385	0.0074	0.2411	0.0041
ψ_{Dec} (Dec. dummy)	0.6716	0.0206	0.6754	0.0188
φ_1 (H-type consumer price sensitivity)	0.0137	0.0005	0.0137	0.0002
φ_2 (L-type consumer price sensitivity)	0.0507	0.0023	0.0523	0.0017
σ_ξ (std of software demand shocks)	2.7394	0.1073	2.7421	0.0500
Block 3: Software Pricing				
c_{Saturn} (unit cost of games for Saturn)	14.6518	0.2166	14.7735	0.2322
c_{PS1} (unit cost of games for PS1)	10.7551	0.1601	10.7306	0.1325
c_{N64} (unit cost of games for N64)	18.4576	0.1629	18.5233	0.1726
σ_ς (std of price error)	0.8786	0.0220	0.8949	0.0154
Block 4: Software Entry				
γ_{Saturn} (mean of entry cost to Saturn)	4.7166	0.0428	4.7292	0.0170
γ_{PS1} (mean of entry cost to PS1)	3.6634	0.0505	3.6999	0.0417
γ_{N64} (mean of entry cost to N64)	4.6130	0.0142	4.6139	0.0086
σ_ν (std of entry cost shocks)	2.4869	0.0119	2.4929	0.0105

Figures: Industrial Description

Figure 1: Positive Indirect Network Effect

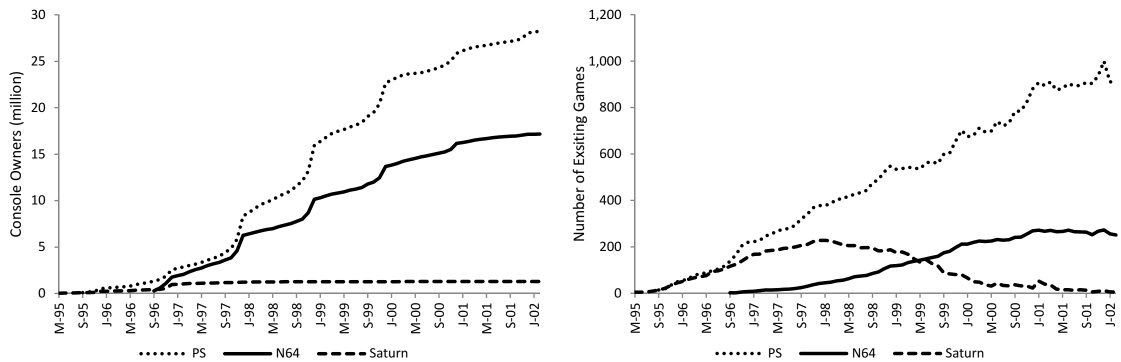
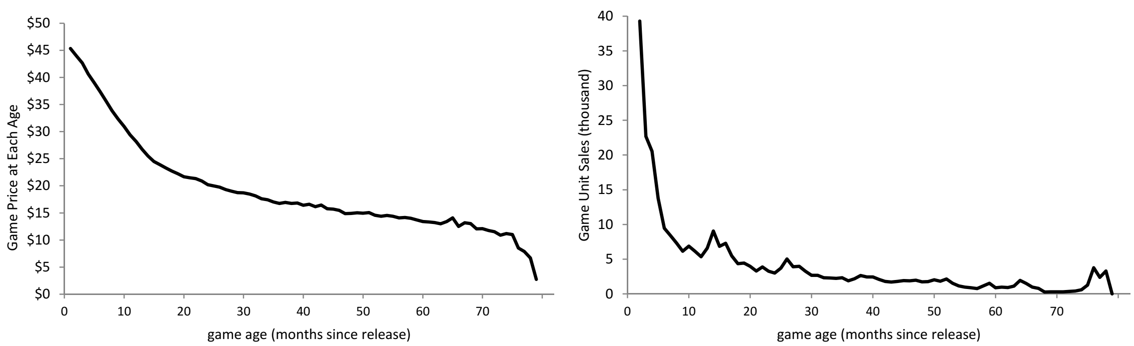


Figure 2: Game Price and Unit Sale at Each Age



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

Figure 3: Seasonality

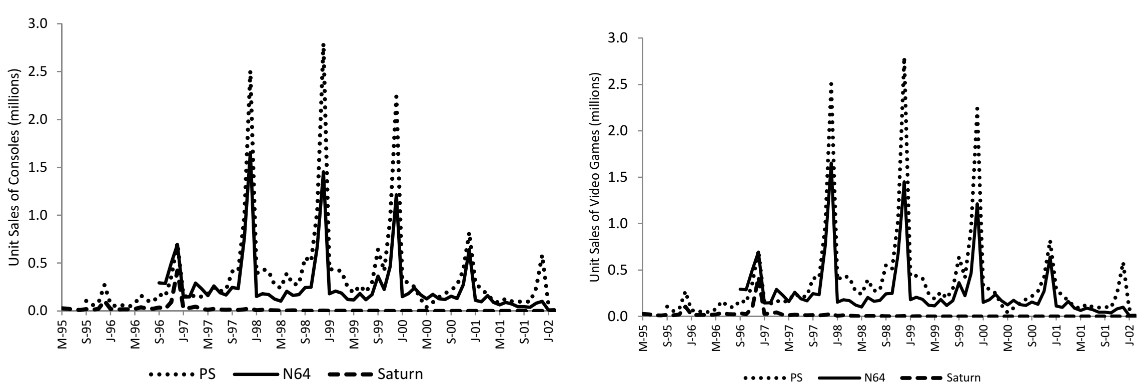
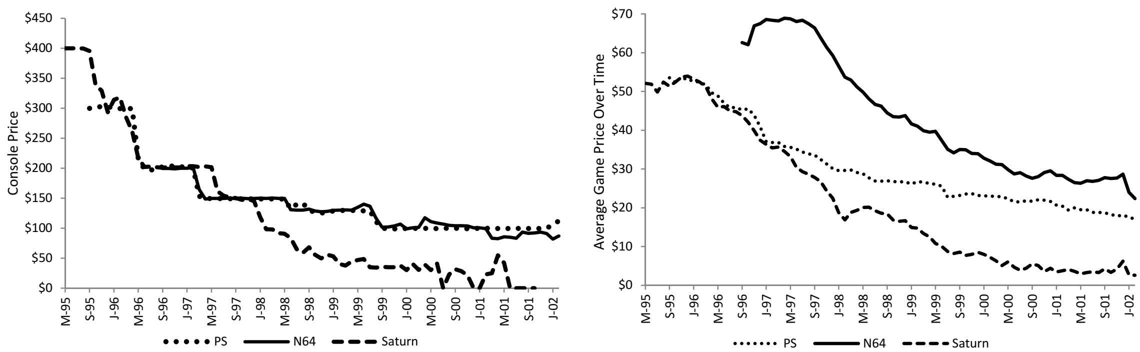
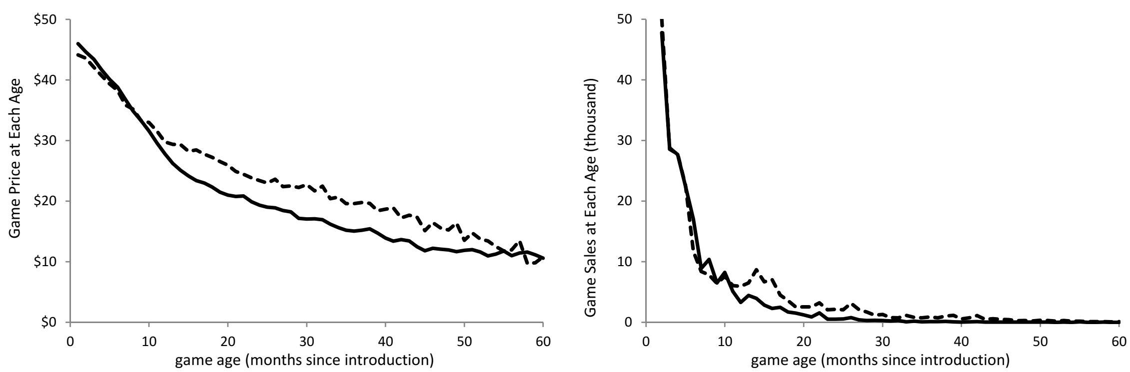


Figure 4: Console and Game Price over Time



Figures: Fit of the Model

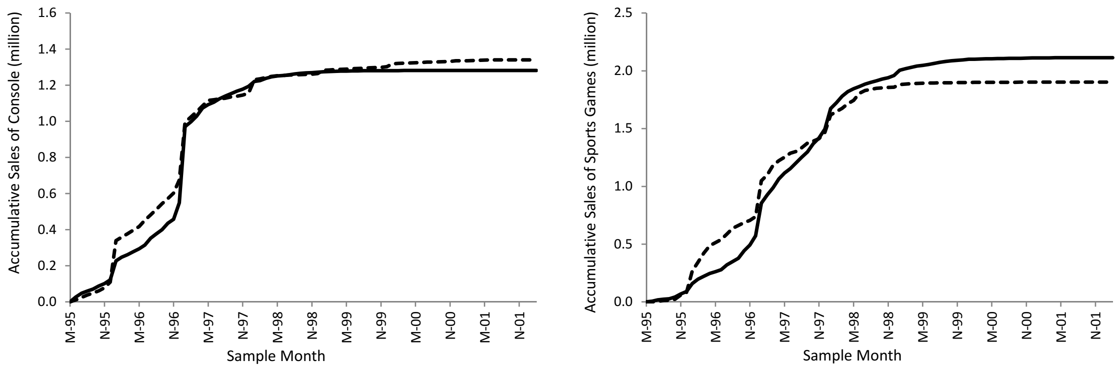
Figure 5: Actual vs. Fitted Game Price and Unit Sales in Age



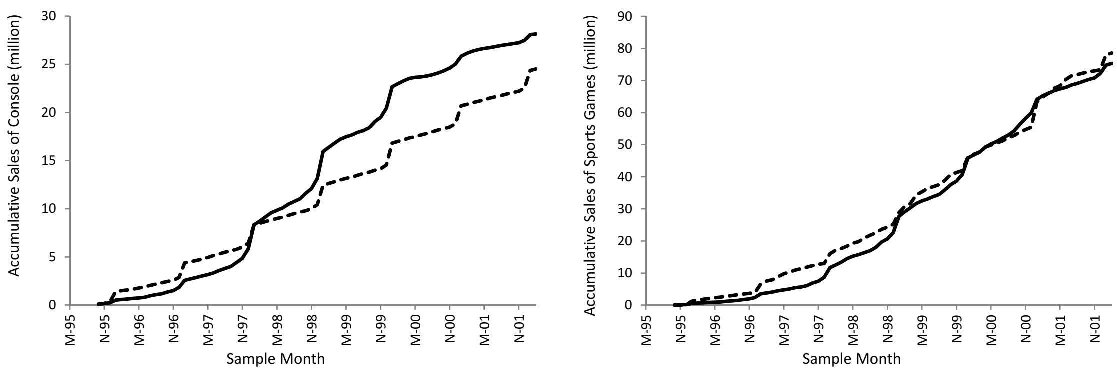
Note: Solid lines represent actual data. Dashed lines are fitted values.

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

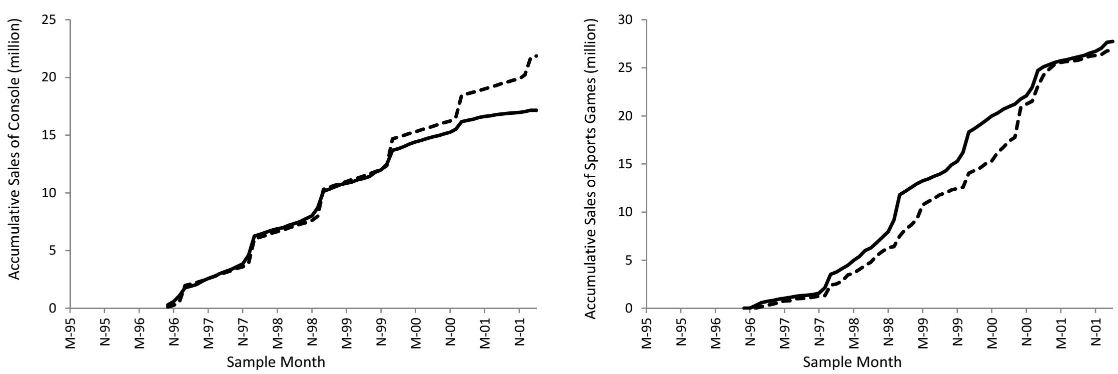
(a) Sega Saturn



(b) Sony PlayStation One



(c) Nintendo 64



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

Appendix A: Estimation Algorithm

Outer-Loop: MCMC

Iteration 0:

- draw $\theta^{(0)}$ frpm prior
- assume myopic consumers and firms, solve the game, obtain
  - $\Psi_{jt}^{(0)}(s_{mt},\theta^{(0)})$ for all m, j, t
  - $V_{rjt}^{(0)}(s_{mt},\theta^{(0)})$ for all m, j, t
  - $G_{it}^{(0)}(s_{mt},p_{mt},\theta^{(0)})$ for all m, j, t

⋮

Iteration k:

- draw $\theta^{*(k)}|\theta^{(k-1)}$
- compute the likelihood $L(data|\theta^{*(k)})$ ⟹
- calculate the acceptance ratio
  - If accept, then $\theta^{(k)} = \theta^{*(k)}$
  - If reject, then $\theta^{(k)} = \theta^{(k-1)}$

⋮

Inner-Loop: Solve the Dynamic Game

(1) approximate equilibrium strategy using pseudo-best responses of past MCMC iterations

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

(2) approximate value functions using pseudo-value functions of past MCMC iterations

$$\hat{V}_{jt}^{(k)}\left(s_{mt},\theta\right) = \sum_{n=1}^{N(k)} V_{jt}^{(k-n)}\left(s_{mt},\theta^{*(k-n)}\right) \times \frac{K_h(\theta-\theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta-\theta^{*(k-n)})}$$

$$\hat{G}_{it}^{(k)}\left(s_{mt},p_{mt},\theta\right) = \sum_{n=1}^{N(k)} G_{it}^{(k-n)}\left(s_{mt},p_{mt},\theta^{*(k-n)}\right) \times \frac{K_h(\theta-\theta^{*(k-n)})}{\sum_{n=1}^{N(k)} K_h(\theta-\theta^{*(k-n)})}$$

(3) interpolate and obtain each agent's continuation value:

$$\hat{E}\left[V_{jt+1}^{(k)}(s_{mt+1}|s_{mt},\hat{\Psi}^{(k)},\theta\right] \text{ and } \hat{E}\left[G_{it+1}^{(k)}(s_{mt+1},p_{mt+1}(s_{mt+1})|s_{mt},\hat{\Psi}^{(k)},\theta\right]$$

(4) solve single-agent DP problem to obtain $\Psi_{jt}^{(k)}(s_{mt},\theta)$

$$V_{jt}^{(k)}(s_{mt},\theta) = \max_{\tilde{p}_{jt}} \pi_{jt}(s_{mt},\tilde{p}_{jt},\hat{\Psi}_{t}^{(k)},\theta) + \beta_f \hat{E}\left[V_{jt+1}^{(k)}(s_{mt+1}|s_{mt},\tilde{p}_{rjt},\hat{\Psi}^{(k)},\theta\right]$$

$$G_{it}^{(k)}(s_{mt},p_{mt},\theta) = \ln\{\sum_{j\in J_{mt}} \exp(X_{jt}\psi - \varphi_i p_{jt} + \xi_{jt}) + \exp\left(\beta_c \hat{E}[G_{it+1}^{(k)}(s_{mt+1},p_{mt+1}(s_{mt+1})|s_{mt},\hat{\Psi}^{(k)},\theta)]\right)\}$$

(5) store pseudo-best response and value functions for future iterations:

$\Psi_{jt}^{(k)}(s_{mt},\theta)$, $V_{jt}^{(k)}(s_{mt},\theta)$ and $G_{it}^{(k)}(s_{mt},p_{mt},\theta)$

Appendix B: Competition Structure of Software Market

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

Table 4: Empirical Results of Testing Software Competition Structure

	Model 1 price ($)	Model 2 $ln(q_{jt})$	Model 3 $ln(q_{jt})$
its own price		-.0126***	-.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

Note: * 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 since release; (iv) the market size measured by the log of the console owners; and (v) monthly dummy. 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 the online rating score, declining in game 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 software price. To address the endogeneity of the price, I use the lagged price as an instrument for the current price. The third column in table 4 lists the estimation results which are consistent with the assumption of strong competition within a market and weak competition across markets.

The last regression mimics Nair (2007).[25] 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.

Appendix C: Computation of the Jacobian Matrix

The Jacobian Matrix in equation (10) is

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

with

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

[25] 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.

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

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

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

Appendix D: Best Response in Price

The 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

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

How to compute the marginal effect of current price on those expected 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: $n_{mit+1} = n_{mit} b_{m_0it} + Q_{mit}$. Hence, the analytical form of $\partial \mathbf{s}_{mt+1}(p_{jt}, p_{-jt}, \mathbf{s}_{mt})/\partial p_{jt}$ is known. Furthermore, along the estimation procedure, I approximated the value functions $V_j(\mathbf{s}_{mt+1})$ by using polynomial regression in state variables. Therefore, I can pin down how current price affects the expectation of incumbent value function. A similar approach can be applied to compute the marginal effect of current price on consumers' continuation values.